

Perpustakaan SKTM

**ANALISIS SUARA:
REKABENTUK PENJANA KOD UNIK**

ASLIZAWANI BINTI ALI

WEK 000327

Dibawah seliaan:

ENCIK ZAIDI BIN RAZAK

Perpustakaan Universiti Malaya



A511275060

ABSTRAK

Satu projek dibangunkan bagi menghasilkan satu kod unik yang boleh menganalisis suara manusia sebagai input. Dengan kata lain kod unik ini boleh membaca suara manusia sebagai input bagi menggantikan kod arahan yang biasanya ditulis dalam bahasa pengaturcaraan tertentu.

Sampel suara direkodkan dan disimpan dengan extension " *.wav " yang merupakan sejenis fail wave. Satu aturcara menggunakan Matlab di tulis untuk membaca data audio dalam sampel yang dibentuk. Kemudian data di analisis menggunakan Fast Fourier Transformation bagi mendapatkan spektrum frekuensi bagi data tersebut.

Data dalam bentuk spektrum frekuensi ini seterusnya akan melalui proses penyahkomposisian dan proses-proses kelulusan dalam wavelet bagi mendapatkan keluaran akhir proses. Berdasarkan keluaran akhir proses wavelet, data yang diperolehi akan di analisis. Hasil keluaran akhir inilah merupakan sumber utama untuk menghasilkan kod unik.

PENGHARGAAN

Syukur Alhamdulillah saya panjatkan ke hadrat Ilahi di atas kurnia-Nya dapat saya menyiapkan tesis ini. Setinggi-tinggi penghargaan kepada Prof. Ir. Dato' Dr. Maskhuri bin Hj Yaacob selaku moderator dan juga buat Encik Zaidi bin Razak selaku penyelia saya yang telah banyak memberi tunjuk ajar serta nasihat kepada saya dalam menyiapkan tesis ini. Segala tunjuk ajar yang diberikan telah banyak memberi saya panduan dalam usaha menyempurnakan tesis ini. Tidak lupa juga kepada semua pensyarah yang telah banyak membimbing saya hingga ke tahap ini.

Ucapan jutaan terima kasih kepada ibu bapa serta keluarga yang sentiasa memberi dorongan dan galakkan untuk saya terus berusaha biar pun jauh di mata. Sesungguhnya tanpa doa dan restu mereka, saya tidak akan sampai ke tahap ini.

Buat rakan-rakan seperjuangan di fakulti mahupun di kolej, terima kasih di atas segala nasihat dan sokongan yang kalian berikan. Walaupun berbeza bidang pengajian namun persahabatan yang terjalin tetap menjadikan kita saling memerlukan.

Sekian, terima kasih.

KANDUNGAN

MUKA SURAT

Abstrak	i
Penghargaan	ii
Isi kandungan	iii
Senarai Jadual	vii
Senarai Rajah	viii

Bab 1 : Pengenalan

1.1 Pengenalan Projek	1
1.2 Objektif Projek	3
1.3 Skop Projek	4
1.4 Spesifikasi Projek	6
1.5 Perancangan Projek	7
1.6 Penjadualan Projek	10
1.7 Ringkasan	12

Bab 2 : Kajian Literasi

2.1 Pengenalan	13
2.2 Kajian Keatas Suara Manusia	14

2.2.1	Pengecaman Suara	15
2.2.2	Spesifikasi dalam Voice Recognition	16
2.2.3	Bagaimana Pengecaman Suara Dilakukan	17
2.3	Kajian Format Fail	20
2.3.1	Jenis Format Fail	21
2.2.1.1	MPEGLayer3	
2.2.1.2	Audio Interchange File Format	
2.2.1.3	Advanced System Format	
2.2.1.4	Wave	
2.2.1.5	Musical Instrument Digital Interface	
2.3.2	Perbandingan	36
2.3.3	Kesimpulan	37
2.3.4	Faktor Pemilihan	42
2.4	Kajian Ke Atas Teknik dan Peralatan Yang Digunakan	44
2.4.1	Wavelet	44
2.4.1.1	Penukarganti Diskrit Wavelet	
2.4.1.2	Algoritma Piramid	
2.4.1.3	Fungsi Kelulusan	
2.4.2	Matlab	49
2.4.3	Fast Fourier Transform	50

Bab 3 : Pelaksanaan Rekabentuk

3.1	Pembentukan Fail Data	52
3.2	Membaca Fail Data	55
3.3	Proses Penyahkomposisian	61
3.4	Pembinaan Semula	64
3.5	Penghasilan Nilai Unik	66
3.6	Ringkasan	67

Bab 4 : Analisis

4.1	Pengenalan	68
4.2	Analisis Ke Atas Nilai (x)	69
4.2.1	Kumpulan Sampel 1	70
4.2.2	Kumpulan Sampel 2	72
4.2.3	Kumpulan Sampel 3	74
4.2.4	Perbandingan Antara Kumpulan Sampel	76
4.2.5	Kesimpulan Daripada Analisis	78
4.3	Ringkasan	80

Bab 5 : Perbincangan

5.1	Faktor Yang memungkinkan Ralat	81
5.2	Perkembangan	84
5.3	Ringkasan	86
Rujukan		87
Lampiran		

SENARAI RAJAH

BIL	NOMBOR RUJUKAN	KETERANGAN	MUKA SURAT
1	1.1	Carta Gantt bagi aktiviti projek	11
2	2.1	Binaan asas bagi voice recognition	16
3	2.2	Gambarajah pengepala (header) bagi fail MP3	23
4	2.3	Binaan asas chunk	26
5	2.4	Struktur chunk dalam fail AIFF	27
6	2.5	Gambarajah format bagi fail ASF	29
7	2.6	Gambarajah format bagi objek data dalam fail	30
8	2.7	format Chunk Pengepala bagi fail MIDI	32
9	2.8	format chunk track bagi fail MIDI	35
10	2.9	struktur chunk dalam fail Wave	38
11	2.10	Gambaran terperinci struktur chunk dalam fail Wave	40
12	2.11	Algoritma piramid	47
13	2.12	Fungsi kelulusan rendah	48
14	2.13	Fungsi kelulusan tinggi	48
15	3.1	Penukaran fail dari perakam suara MP3 kepada bentuk wav	52
16	3.2	Paparan output dari <i>header</i> fail	57
17	3.3	Isyarat yang terbentuk daripada suara yang telah direkodkan	58
18	3.4	Spektrum frekuensi yang terbentuk	58
19	3.5	graf bagi nilai spektrum frekuensi	60
20	3.6	proses penyahkomposisian dan pembinaan semula dalam wavelet.	66
21	4.1	Perbandingan purata nilai (x) bagi setiap sampel dalam kumpulan 1 dengan fail 'open'	71

BIL	NOMBOR RUJUKAN	KETERANGAN	MUKA SURAT
22	4.2	Peratusan Ralat yang wujud dalam setiap sampel kumpulan satu.	71
23	4.3	Perbandingan purata nilai (\bar{x}) bagi setiap sampel dalam kumpulan 2 dengan fail 'open'	73
24	4.4	Peratusan Ralat yang wujud dalam setiap sampel kumpulan 2	73
25	4.5	Perbandingan purata nilai (\bar{x}) bagi setiap sampel dalam kumpulan 3 dengan fail 'open'	75
26	4.6	Peratusan Ralat yang wujud dalam setiap sampel kumpulan 3	75
27	4.7	Peratusan Ralat yang wujud dalam ketiga-tiga kumpulan sampel	76
28	4.8	Perbandingan purata nilai (\bar{x}) bagi ketiga tiga kumpulan sampel	77
29	5.1	Perbandingan purata nilai (s) dan (s_{rec}) bagi setiap sampel dalam kumpulan satu.	82
30	5.2	Peratusan Ralat yang wujud dalam kumpulan Sampel 1	83

SENARAI JADUAL

BIL	NOMBOR RUJUKAN	KETERANGAN	MUKA SURAT
1	1.1	Perancangan awal proses pembangunan	9
2	1.2	Penjadualan aktiviti pembangunan projek	10
3	2.1	Penerangan mengenai pengepala fail MP3	23
4	2.2	Penerangan mengenai struktur chunk dalam fail AIFF	27
5	2.3	Penerangan mengenai struktur chunk pengepala dalam fail MIDI	33
6	2.4	Penerangan mengenai chunk track	35
7	2.5	Perbandingan antara beberapa jenis format fail	36
8	2.6	Penerangan struktur chunk dalam fail wave	41
9	3.1	Maklumat dari header fail data	55
10	4.1	Purata nilai unik bagi setiap sampel kumpulan 1 serta ralat yang wujud	70
11	4.2	Purata nilai unik bagi setiap sampel kumpulan 2 serta ralat yang wujud	72
12	4.3	Purata nilai unik bagi setiap sampel kumpulan 3 serta ralat yang wujud	74
13	4.4	purata ralat bagi ketiga tiga kumpulan sampel	76
14	5.1	Perbezaan antara nilai s dan s_rec dalam sampel kumpulan satu serta purata nilai ralat yang muncul	81

Dunia pada hari ini tidak lagi seperti dulu. Kemudahan teknologi komunikasi membuat kita bisa terhubung dengan siapa saja kapan saja. Perkembangan teknologi ini membawa kita ke era yang baru, di mana kita bisa melakukan banyak hal yang sebelumnya tidak mungkin. Kita dapat berkomunikasi dengan orang-orang di seluruh dunia hanya dengan beberapa klik mouse.

Bab 1

Pengenalan

1.1 PENGENALAN PROJEK

Dunia pada hari ini tidak lagi seperti dulu. Kemunculan pelbagai teknologi canggih bukan lagi satu perkara asing bagi penduduk dunia. Perkembangan teknologi hari ini walaupun ada baik dan buruknya, tetapi tidak dapat dinafikan bahawa ianya banyak memberi kemudahan kepada kita dalam menjalankan kegiatan seharian.

Kemunculan Internet telah mewujudkan jaringan maklumat yang meluas manakala teknologi ICT pula membawa dunia kepada era globalisasi di mana manusia boleh berkomunikasi tanpa mengira sempadan. Melihat kepada perkembangan teknologi pada hari ini tidak mustahil suatu hari nanti semua aktiviti manusia digantikan oleh mesin-mesin yang dicipta khusus untuk tujuan tersebut. Cuba bayangkan andainya kita tidak perlu lagi menggunakan kunci rumah untuk membuka pintu tetapi sebaliknya hanya dengan mengucapkan perkataan seperti “ open “ dan pintu akan terus terbuka.

Projek ini antara lainnya cuba melakukan kajian dan penyelidikan untuk menjadikan senario tersebut sebagai satu realiti. Secara keseluruhannya projek ini cuba mencari keunikan pada suara individu untuk menghasilkan satu algoritma yang boleh membaca dan menganalisis suara manusia sebagai satu arahan menggantikan kod-kod arahan yang biasanya ditulis dalam bahasa pengaturcaraan tertentu.

1.2 Apa yang dilakukan adalah dengan merekodkan suara sebagai satu sampel data audio. Sampel data ini di analisis untuk mendapatkan data yang lebih terperinci terutamanya amplitud dan frekuensinya. Data-data yang diperolehi diprogramkan dengan menggunakan bahasa pengaturcaraan yang bersesuaian. Sila rujuk lampiran A untuk gambaran yang lebih jelas mengenai teknologi ini.

1.2 OBJEKTIF PROJEK

Projek ini telah menggariskan beberapa objektif penting yang perlu di capai bagi memastikan kejayaannya. Antara objektif tersebut termasuklah :

- 1.2.1 Membentuk algoritma yang boleh membaca dan menganalisis data audio
- 1.2.2 Menggantikan kod-kod arahan dengan suara manusia
- 1.2.3 Mewujudkan persekitaran kehidupan yang elektronik sejajar dengan perkembangan teknologi semasa.

1.3 SKOP PROJEK

1.3.1 Pembentukan kod unik

Matlamat utama projek ini adalah untuk menghasilkan suatu kod unik yang boleh menggantikan kod arahan dengan suara manusia. Jadi apa yang penting adalah mendapatkan data-data yang unik mengenai suara individu yang membezakan suara seseorang individu dengan individu yang lain. Kajian terperinci perlu dilakukan untuk mendapatkan data data yang tepat.

1.3.2 Penggunaan suara sebagai data audio

Data audio terdiri daripada pelbagai jenis sama ada musik, lagu dan sebagainya. Apa yang ditekankan dalam projek ini adalah data audio yang digunakan adalah suara manusia. Suara seseorang individu akan direkod dan di analisis untuk mendapat data yang lebih terperinci.

1.3.3 *Speaker Dependent*

Kajian atau kod yang bakal dihasilkan adalah berkonsepkan Speaker Dependent iaitu pengecaman suara individu tertentu sahaja. Maka

satu kaedah diperlukan untuk memastikan sistem dapat mengenali suara individu tersebut

1.3.4 *Text Independent*

Individu terbabit bebas menyebut apa sahaja perkataan.

1.4 SPESIFIKASI PROJEK

Spesifikasi projek adalah salah satu langkah yang di ambil untuk mengenal pasti segala peralatan atau perkakasan dan juga perisian yang digunakan untuk menjayakan projek. Selain daripada keperluan asas seperti komputer peribadi, pencetak, sistem pengendalian dan sebagainya, antara perkakasan dan perisian lain yang digunakan dalam projek ini adalah seperti berikut :

1.4.1 Perkakasan

- Perakam suara MP3

1.4.2 Perisian

- MatLab
- M-Pack 1
- Microsoft Visual C++

1.5 PERANCANGAN PROJEK

Bagi memastikan projek berjalan dengan lancar, perancangan awal adalah penting. Perancangan awal yang baik mampu memberi sedikit sebanyak gambaran kepada pembangun projek tentang perjalanan keseluruhan projek untuk memastikan gerak kerja tidak tertangguh. Jadual 1.1 menunjukkan perancangan awal mengenai gerak kerja projek secara keseluruhannya.

FASA	GERAK KERJA
Perancangan projek	<ul style="list-style-type: none">▪ Membuat kajian terhadap tajuk projek yang dipilih.▪ Membuat sedikit sebanyak perancangan mengenai projek yang akan dijalankan .▪ Mengenalpasti fungsi-fungsi serta data-data penting yang perlu dikumpulkan.▪ Mengkaji perkakasan dan perisian yang bersesuaian untuk digunakan.
Analisa	<ul style="list-style-type: none">▪ Mula membuat kajian untuk mengunpulkan data data yang diperlukan▪ Membuat analisa mengenai format-format fail

	<p>bagi penyimpanan audio.</p> <ul style="list-style-type: none"> ▪ Mengkaji keunikan dalam suara manusia ▪ Membuat analisa mengenai amplitud dan frekuensi dalam bunyi ▪ Mengetahui kaedah-kaedah untuk memproses dan menganalisis audio data. ▪ Mengkaji bagaimana suara manusia boleh direkodkan sebagai audio data. ▪ Mendapatkan contoh-contoh pengaturcaraan bunyi bagi mengenal pasti bahasa pengaturcaraan yang sesuai. ▪ Mengkaji platform yang bersesuaian
Rekabentuk	<ul style="list-style-type: none"> ▪ Mengumpulkan data – data yang diperolehi dari fasa analisis. ▪ Menentukan format fail yang akan digunakan ▪ Menentukan bahasa pengaturcaraan dan platform yang akan digunakan. ▪ Menghuraikan fungsi algoritma yang dibentuk dalam bentuk gambarajah

implimentasi	<ul style="list-style-type: none"> ▪ Memulakan proses menghasilkan algoritma ▪ dengan merekodkan suara dan menyimpan data mengikut format yang dipilih. ▪ Memulakan proses menulis aturcara pada platform yang telah dipilih.
Pengujian	<ul style="list-style-type: none"> ▪ Menguji keberkesanan aturcara yang telah ditulis ▪ Mengenal pasti kelemahan-kelemahan dalam aturcara yang ditulis ▪ Memperbaiki segala kelemahan yang timbul dalam aturcara
Dokumentasi	<ul style="list-style-type: none"> ▪ Proses merekodkan segala data-data dan maklumat dari peringkat awal proses pembangunan projek hingga ke proses pengujian.

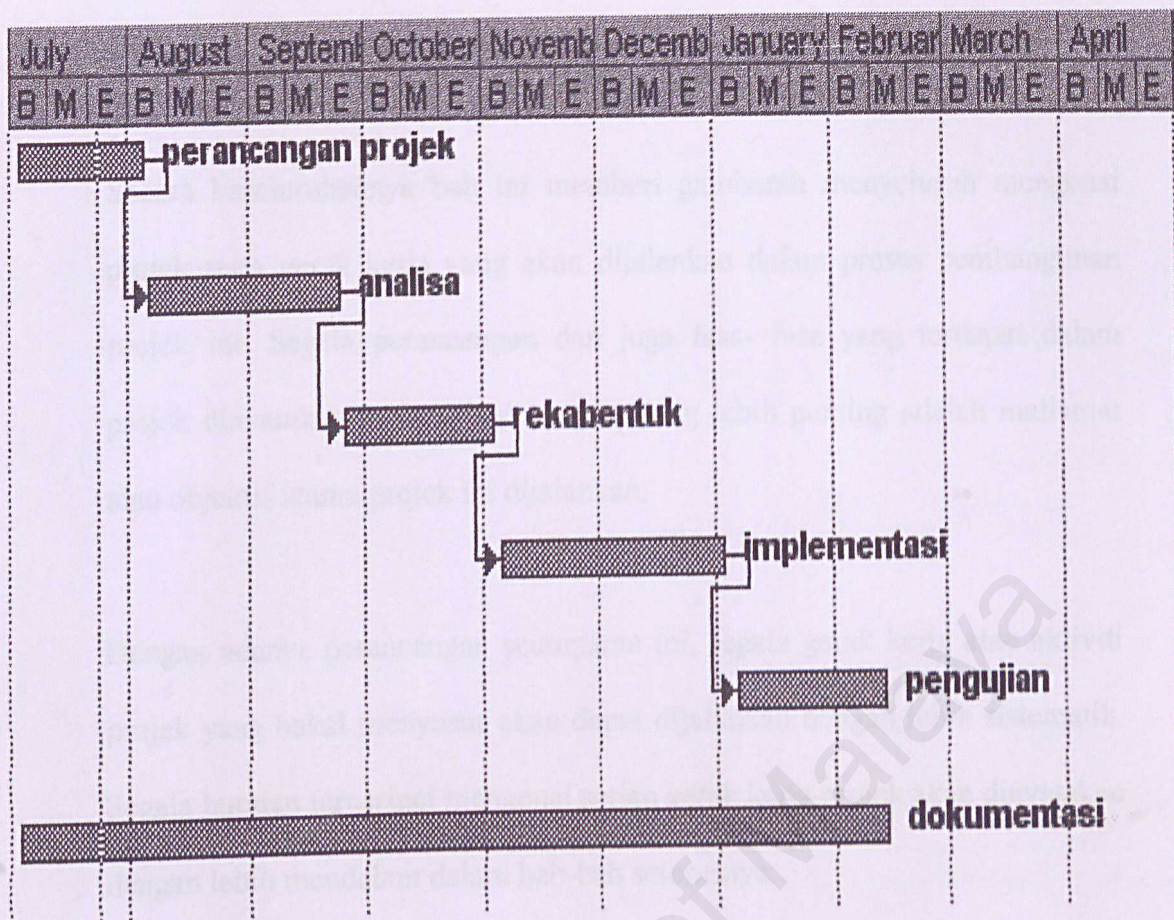
Jadual 1.1 : Perancangan awal proses pembangunan projek

1.6 **PENJADUALAN PROJEK**

Penjadualan projek juga merupakan salah satu aspek penting dalam pembangunan projek yang tidak boleh dianggap remeh. Penjadualan amat penting bagi memastikan projek dapat disiapkan pada masa yang telah ditetapkan. Jadual 1.2 dan rajah 1.1 menunjukkan penjadualan bagi keseluruhan aktiviti projek.

Task Name	Duration	Start	Finish
perancangan projek	23 days	Thu 03-07-03	Mon 04-08-03
analisa	36 days	Wed 06-08-03	Wed 24-09-03
rekabentuk	27 days	Fri 26-09-03	Mon 03-11-03
implementasi	43 days	Wed 05-11-03	Fri 02-01-04
pengujian	29 days	Tue 06-01-04	Fri 13-02-04
dokumentasi	162 days	Thu 03-07-03	Fri 13-02-04

Jadual 1.2 : Penjadualan aktiviti pembangunan projek



Rajah 1.1 : Carta Gantt bagi aktiviti projek

1.7 RINGKASAN

Secara keseluruhannya bab ini memberi gambaran menyeluruh mengenai projek serta gerak kerja yang akan dijalankan dalam proses pembangunan projek ini. Segala perancangan dan juga fasa- fasa yang terdapat dalam projek dinyatakan secara ringkas .Apa yang lebih penting adalah matlamat atau objektif utama projek ini dijalankan.

Dengan adanya perancangan seumpama ini, segala gerak kerja atau aktiviti projek yang bakal menyusur akan dapat dijalankan dengan lebih sistematik. Segala huraian terperinci mengenai setiap gerak kerja projek akan dinyatakan dengan lebih mendalam dalam bab-bab seterusnya.

Kajian literasi merupakan kegiatan yang penting dalam proses pembelajaran untuk meningkatkan kemampuan literasi siswa. Kegiatan ini bertujuan untuk meningkatkan kemampuan literasi siswa yang akan membantu mereka dalam memahami dan menggunakan informasi yang ada di sekitar mereka.

Tujuan utama kajian literasi adalah untuk meningkatkan kemampuan literasi siswa yang akan membantu mereka dalam memahami dan menggunakan informasi yang ada di sekitar mereka. Kegiatan ini bertujuan untuk meningkatkan kemampuan literasi siswa yang akan membantu mereka dalam memahami dan menggunakan informasi yang ada di sekitar mereka.

Bab 2

Kajian Literasi

2.1 PENGENALAN

Kajian literasi merupakan bahagian yang penting dalam proses pembangunan projek di mana pada peringkat ini segala maklumat dan keperluan projek dikenalpasti. Kajian yang dilakukan adalah secara menyeluruh dan mengambil kira semua aspek yang berkaitan.

Tujuan utama kajian literasi dijalankan adalah untuk mendapatkan segala maklumat yang diperlukan untuk membangunkan projek. Di samping itu, segala ketidakpastian yang timbul mengenai maklumat yang telah diperolehi di kaji untuk mendapatkan kesahihannya. Ini adalah penting bagi memastikan maklumat yang diperolehi adalah tepat.

Selain itu kajian literasi dijalankan untuk mengkaji dan membuat penilaian awal terhadap projek yang bakal dibangunkan. Ini sedikit sebanyak dapat memberi gambaran awal serta membantu memberi pemahaman yang lebih jelas terhadap projek tersebut.

Antara lainnya kajian literasi untuk projek ini merangkumi kajian terhadap jenis-jenis fail bagi muzik, format data yang digunakan oleh setiap jenis fail serta kajian ke atas keunikan pada suara manusia yang akan dibincangkan dengan lebih lanjut dalam bahagian berikutnya.

2.2 KAJIAN KE ATAS SUARA MANUSIA

Setiap individu mempunyai suara yang berbeza daripada individu yang lain. Secara amnya suara manusia dibezakan oleh faktor-faktor seperti[6]:

- a) Jantina sama ada lelaki atau perempuan
- b) Umur sama ada dewasa atau kanak-kanak
- c) Loghat percakapan mengikut kawasan tempat tinggal.

Namun begitu, masih terdapat faktor-faktor lain yang membezakan suara manusia yang tidak dapat dilihat dengan jelas. Oleh yang demikian kajian ini dijalankan untuk mengkaji suara manusia dari segi fizikalnya dengan lebih mendalam lagi. Antara objektif yang ingin dicapai melalui kajian ini termasuklah :

- a) Mengesan keunikan dalam suara manusia contohnya mungkin dari segi frekuensi ataupun amplitud
- b) Mengenal pasti kaedah bagi menganalisis suara manusia

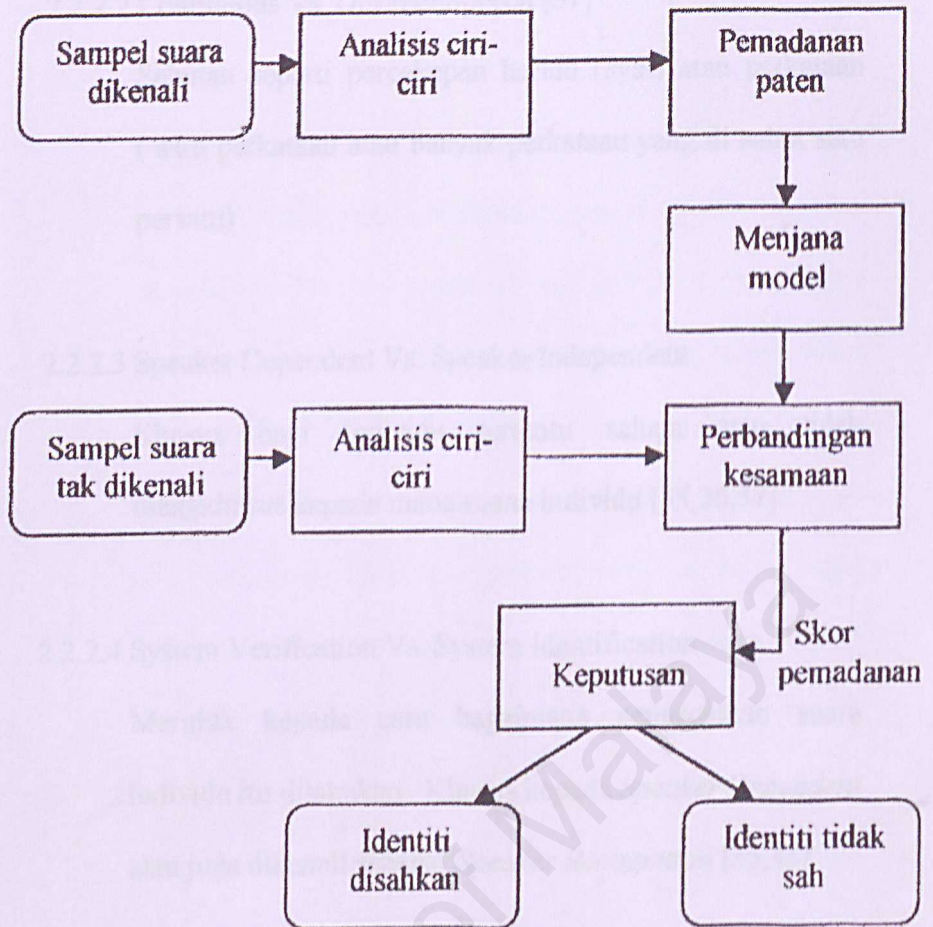
Di samping itu kajian ini juga turut merangkumi kaedah-kaedah untuk merekodkan suara manusia.

2.2.1 Pengecaman suara (*voice recognition*)

Pengecaman suara adalah satu teknologi baru di mana bunyi, perkataan atau pernyataan yang di sebut oleh manusia ditukarkan kepada isyarat elektrik. Isyarat ini kemudiannya ditukarkan ke dalam bentuk atau corak pengkodan yang telah dikenalpasti. Secara lebih jelas lagi teknologi ini cuba menjadikan suara manusia sebagai input bagi menggantikan kod-kod arahan yang biasanya digunakan dalam sesebuah sistem berkomputer.

Fokus diberikan kepada suara manusia kerana kita seringkali menggunakan suara untuk berkomunikasi. Walau bagaimanapun penggunaan suara manusia menjadi sukar kerana setiap manusia mempunyai percakapan yang berbeza-beza. Adakalanya pecakapan bergantung kepada persekitaran serta emosi setiap individu. Ini menyebabkan ciri-ciri atau keunikan suara seseorang individu itu sukar dikenalpasti.

Sebaliknya program komputer biasanya direkabentuk untuk memberikan tindak balas bergantung kepada input yang betul-betul tepat. Oleh yang demikian hasil yang tepat daripada proses analisis ke atas suara adalah sangat penting. Rajah dibawah menunjukkan binaan asas bagi pengecaman suara [39].



Rajah 2.1 Binaan asas bagi voice recognition

2.2.2 Spesifikasi dalam *voice recognition*

2.2.2.1 Arahan Vs. Tulisan

Sama ada perkataan yang disebut berupa arahan atau untuk dipaparkan pada skrin

2.2.2.2 Continuous Vs. Discrete Speech [37]

Sebutan seperti percakapan harian (ayat) atau perkataan (satu perkataan atau banyak perkataan yang di sebut satu persatu)

2.2.2.3 Speaker Dependent Vs. Speaker Independent

Khusus bagi individu tertentu sahaja atau tidak mengkhusus kepada mana-mana individu [35,36,37]

2.2.2.4 System Verification Vs. System Identification

Merujuk kepada cara bagaimana pengecaman suara individu itu dilakukan . Khusus kepada *Speaker Dependent* atau juga dikenali sebagai *Speaker Recognition* [35,38]

2.2.2.5 Text Dependent Vs. Text Independent

Khusus bagi perkataan tertentu sahaja ataupun pengguna bebas menyebut sebarang perkataan. [35,36,38]

2.2.3 Bagaimana pengecaman suara dilakukan.

Terdapat banyak pendekatan yang boleh digunakan untuk mengimplementasikan pengecaman suara. Pendekatan yang paling biasa digunakan adalah dengan membahagikan proses kepada dua bahagian utama seperti berikut [3] :

a) Analisis ciri-ciri (*feature analysis*)

Memandangkan setiap orang mempunyai corak percakapan berbeza, sistem perlu terlebih dahulu mempelajari corak percakapan dan mengenalpasti ciri-ciri yang unik dalam suara individu tersebut. Melalui kaedah ini, sistem akan mempamerkan beberapa perkataan yang perlu di sebut oleh pengguna. Pada masa yang sama sistem cuba mendapatkan nilai purata bagi setiap sampel yang di sebut oleh pengguna dan membentuk satu template yang mengandungi corak percakapan individu tersebut.

Walau bagaimanapun cara ini terhad oleh beberapa perkataan yang di sebut oleh pengguna semasa sesi pembelajaran oleh sistem sahaja. Selain itu, ia juga terhad kepada pengguna yang telah pun menjalani sesi pembelajaran tersebut.

Satu lagi kaedah yang lebih berkesan bagi mendapatkan corak input yang sama atau hampir sama dengan corak atau bentuk yang telah dikenalpasti oleh program komputer ialah dengan melakukan satu analisis menggunakan pendekatan-pendekatan tertentu. Dua pendekatan yang biasa digunakan adalah *Fourier Transforms* dan juga *Linear Predictive Coding*. Pendekatan ini digunakan untuk

mengenalpasti ciri-ciri unik dan mengesan persamaan antara input dan juga data suara sebenar .

b) Pemadanan corak atau bentuk (*pattern matching*)

Pemadanan corak merupakan proses mengenalpasti input suara yang dikeluarkan oleh program dengan membandingkannya dengan template atau corak yang sudah difahami oleh program komputer.

2.3 KAJIAN FORMAT FAIL

Kajian yang pertama ini merupakan kajian keatas fail-fail muzik. Antara maklumat yang perlu dikenalpasti atau objektif bagi kajian ini termasuklah :

- a) Mengenalpasti fail-fail muzik yang ada
- b) Mendapatkan format atau cara penyimpanan data bagi setiap jenis fail
- c) Memilih format fail yang bersesuaian.

Langkah pertama dalam menjalankan kajian adalah mengenal pasti jenis-jenis fail muzik yang ada seperti Musical Instrument Digital Interface (MIDI), Moving Picture Expert Group (M-PEG), Advanced System Format (ASF) dan sebagainya. Kajian yang lebih mendalam dilakukan ke atas setiap fail bagi mengenal pasti struktur atau format data yang disimpan. Maklumat yang dikaji termasuklah :

- a) Jenis data sama ada boleh *di compress* atau tidak
- b) Jenis sistem pengendalian yang menyokong
- c) Jenis channel yang digunakan
- d) Saiz bagi data
- e) Jenis *extension* bagi setiap fail
- f) Kelajuan bagi setiap jenis fail
- g) Saiz fail

h) Bilangan bit per sample

i) Kadar persampelan

Seterusnya kelebihan dan kekurangan setiap jenis fail di kenal pasti. Maklumat-maklumat ini kemudian dikumpul dan di analisa bagi menentukan jenis fail yang paling sesuai dan berkualiti untuk digunakan dalam projek ini.

Kajian ini juga secara tidak langsung cuba mengenal pasti langkah-langkah untuk menghasilkan setiap jenis fail serta ciri-ciri lain yang membezakan satu jenis fail dari fail yang lain.

2.3.1 Jenis format fail

Untuk mendapatkan format fail yang paling sesuai untuk digunakan dalam projek ini kajian dibuat ke atas beberapa jenis format yang ada.

2.3.1.1 MPEG Layer 3 (MP3)

MPEG merupakan singkatan bagi *Moving Picture Expert Group*. [1,10,11,12,13,14,15] Jenis fail ini menjadi semakin popular dewasa ini memandangkan saiz failnya yang lebih kecil kerana ciri-ciri pemadatan (*compression*) yang tinggi. Fail ini dikatakan mampu menyimpan data audio

selama 30 minit dalam fail sebesar 4MB pada kadar 16kbps [17]. Fail ini mempunyai extension “*.mp3” bagi setiap nama fail.

Struktur fail MP3 terdiri daripada blok-blok kecil yang di panggil *frames*. Setiap *frames* terdiri daripada pengepala frame (*frames header*) dan diikuti oleh data audio. Terdapat dua jenis format utama bagi fail MP3 iaitu kadar bit yang tetap (*constant bitrate – CBR*) dan kadar bit yang berubah-ubah (*variable bitrate - VBR*). Bagi format CBR kadar bit bagi semua *frame* adalah sama kecuali pada bahagian audio data. Sebaliknya bagi format VBR, nilai ini boleh berbeza dari satu *frame* ke *frame* yang lain.

VBR menjadikan saiz fail lebih kecil kerana ia mengurangkan ruang untuk nada yang rendah dan memperuntukkan ruang yang lebih untuk nada yang lebih tinggi. Strukturnya sama seperti CBR tetapi *frame* pertamanya tidak mempunyai data audio.[11]



Rajah 2.2 : Gambarajah pengepala (*header*) bagi fail MP3

Penerangan lanjut mengenai pengepala fail MP3 ini adalah seperti yang ditunjukkan dalam jadual berikut :

Simbol	Keterangan
A	<ul style="list-style-type: none"> - bit-bit ini bertindak sebagai penyelarass frame - kesemua bit disetkan kepada “1” - sebagai tanda permulaan bagi frame
B	<ul style="list-style-type: none"> - ID bagi versi MPEG - 00 bagi MPEG versi 2.5 - 10 bagi MPEG versi 2 - 11 bagi MPEG versi 1 - 01 bagi bit simpanan - nilai biasa bagi fail MP3 adalah “11”
C	<ul style="list-style-type: none"> - lapisan MPEG yang digunakan - 00 adalah bit simpanan - 01 bagi lapisan III - 10 bagi lapisan II - 11 bagi lapisan I

-
- dalam kebanyakan fail MP3 nilai ini adalah "01"
-

- D
- perlindungan oleh CRC iaitu sejenis mekanisme pemeriksaan sepanjang 16 bit yang diletakkan di antara pengkepala dan data audio.
 - 0 bagi frame yang dilindungi oleh mekanisme ini
 - 1 bagi yang tidak mempunyai mekanisme CRC
 - kebanyakan fail MP3 tidak mempunyai mekanisme ini.
-

- E
- mewakili kadar bit bagi setiap frame.
 - Kadar bit ini mungkin sama bagi setiap frame atau sebaliknya.
 - Mempunyai perwakilan binari tertentu
- seperti nilai binari 0001 mewakili kadar bit 32 kbps.
-

- F
- frekuensi bagi kadar persampelan
 - 00 mewakili 44100 Hz
 - 01 mewakili 48000 Hz
 - 10 mewakili 32000 Hz
 - 11 adalah bit simpanan
 - bagi kebanyakan fail MP3 frekuensi kadar persampelan adalah 44100 Hz.
-

- G
- *padding*
 - digunakan sebagai lapisan tambahan bagi menjadikan kadar aliran data adalah lebih tepat.
 - 0 bagi *frame* yang mempunyai *padding*
 - 1 bagi sebaliknya
-

- H
- bit *private*
 - bit ini digunakan untuk keperluan yang spesifik contohnya bagi melaksanakan aplikasi bagi *event* tertentu.
-

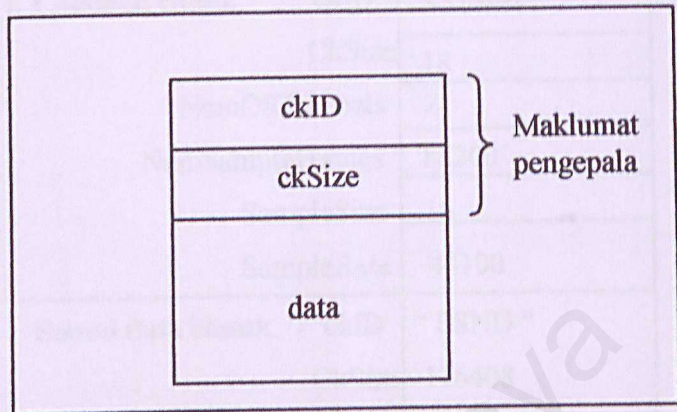
I	- pemilihan <i>channel</i> sama ada stereo, mono, dual ataupun joint stereo yang masing-masing mempunyai perwakilan binari tertentu.
J	- mewakili mod penyambungan (<i>extension mode</i>) bagi <i>frame</i> yang menggunakan <i>channel joint stereo</i> sahaja.
K	- Hakcipta - 0 bagi fail yang tidak dilindungi oleh hakcipta - 1 bagi fail yang dilindungi
L	- original - 0 mewakili fail yang merupakan salinan bagi media asal - 1 mewakili media asal
M	- kenalpasti sama ada terdapat frekuensi yang dimampatkan melebihi cca 3.2 kHz

Jadual 2.1 : Penerangan mengenai pengepala fail MP3

2.3.1.2 Audio Interchange Fail Format (AIFF)

AIFF merupakan jenis fail yang biasa digunakan dalam sistem pengoperasian Macintosh an mempunyai extension “*.aif”. Fail ini digunakan untuk menyimpan data audio dengan pelbagai jenis channel.[1,5,16,18,19,23]

Struktur binaan fail ini adalah dalam bentuk “chunk” di mana bahagian bahagian data tertentu dipecahkan mengikut struktur yang dikenali sebagai “chunk”. Binaan asas chunk adalah seperti berikut [19,23]:



Rajah 2.3 : Binaan asas chunk

Binaan chunk ini terdiri daripada chunk utama iaitu FORM AIFF dan juga local chunk. Local chunk terdiri daripada pelbagai jenis dan kesemuanya adalah sub chunk bagi chunk FORM AIFF. Local chunk yang paling biasa digunakan dan terdapat dalam semua fail AIFF adalah *Common chunk* dan juga *Sound data chunk*. Gambarajah dan jadual di bawah memberikan gambaran yang lebih jelas mengenai struktur fail ini [23].

FORM AIFF

		“ FORM “		
		176516		
		“ AIFF ”		
Common chunk	ckID	“ COMM ”		
	CkSize	18		
	NumOfChannels	2		
	NumSampleFrames	88200		
	SampleSize	16		
	SampleRate	44100		
Sound data chunk	ckID	“ SSND ”		
	CkSize	176408		
	offset	0		
	blockSize	0		
	soundData	Ch1	Ch 2

Rajah 2.4 : struktur chunk dalam fail AIFF

Term	Keterangan
ckID	<ul style="list-style-type: none"> - nilai yang tetap bagi setiap jenis chunk. - unik bagi setiap chunk
ckSize	<ul style="list-style-type: none"> - saiz bagi chunk tertentu - tidak termasuk 8 bytes yang digunakan oleh ckID dan ckSize
NumOfChannels	<ul style="list-style-type: none"> - jenis <i>channel</i> untuk data audio - 1 mewakili mono - 2 mewakili stereo - 4 mewwakili multi channel.
NumSampleFrames	<ul style="list-style-type: none"> - merujuk kepada bilangan sampel frames dalam sound data chunk.

SampleSize	- bilangan bit bagi setiap sample point. - mengambil mana-mana nilai dari 1 hingga 32
SampleRate	- bilangan sampel frames yang dapat dimainkan dalam masa satu saat.
Offset	- tentukan di mana sampel frames pertama bermula
BlockSize	- digunakan bersama offset.
SoundData	- mengandungi sampel frames yang menghasilkan audio

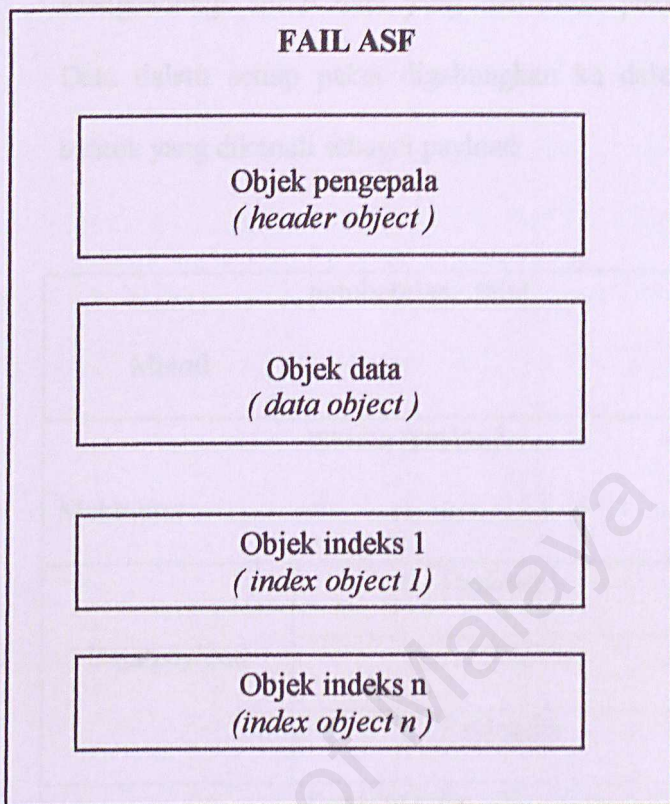
Jadual 2.2 : Penerangan mengenai struktur chunk dalam fail AIFF

2.3.1.3 Advance Systems Format (ASF)

Format ini merupakan salah satu format bagi fail Window Media dengan *extension* .asf. Fail ini menyimpan data-data audio dan video yang boleh dimampatkan dengan menggunakan pelbagai jenis *codec* [1,15,16,31,32]. Saiz failnya adalah lebih kecil berbanding MP3 kerana ciri-ciri pemampatan data yang lebih tinggi. Ia dikatakan mampu menyimpan data audio selama 30 minit dalam fail bersaiz 2MB dengan kelajuan 10kbps.[17]

Setiap fail ASF dibahagikan kepada beberapa seksyen yang dikenali sebagai objek. Terdapat tiga peringkat objek

yang utama seperti yang digambarkan dalam rajah di bawah.[32]



Rajah 2.5 : Gambarajah format bagi fail ASF

a) Objek pengepala (*header object*)

Merupakan peringkat paling tinggi. Ia memuatkan maklumat-maklumat umum mengenai sesuatu fail. Ini termasuklah saiz fail, metod untuk pembetulan ralat dan juga jenis codec yang digunakan.

b) Objek data (*data object*)

Mengandungi aliran data yang berbentuk paket.
Data dalam setiap paket digabungkan ke dalam
bentuk yang dikenali sebagai payload.

pembetulan ralat	
Metod	
umum payload	
Maklumat	
Data payload	Payload 1

	Payload n
Padding data	

Rajah 2.6 : Gambarajah format bagi objek data
dalam fail ASF

c) Objek indeks (*index object*)

Objek indeks ini mengandungi senarai nama paket
serta nombor bagi setiap paket. Melalui senarai ini
aplikasi boleh mengira nombor paket bagi
menentukan paket mana yang bertanggungjawab
untuk persembahan tertentu.

2.3.1.4 Wave

Fail jenis ini mempunyai extension “*.wav” dan biasa digunakan dalam sistem pengoperasian Windows [1,16,20,21,22,28,29,30,33]. Saiznya adalah sangat besar dan biasanya tidak digunakan dalam internet melainkan bagi program yang kecil sahaja iaitu tidak melebihi beberapa minit pada kadar 11 kHz, 16 bit [17].

Struktur fail ini menggunakan struktur RIFF (Resource Interchange File Format) yang juga memecahkan kandungan fail kepada satu struktur yang lebih kecil yang dipanggil chunk. Seperti fail AIFF, chunk juga merupakan binaan asas bagi struktur RIFF. Setiap chunk mempunyai header dan data sendiri. Header chunk mengenalpasti jenis dan saiz bagi bait-bait dalam chunk data. Sesetengah chunk pula mungkin mempunyai sub chunk [20,30,34].

Seperti fail AIFF, fail wave juga terdiri daripada satu chunk utama yang menggunakan ID “RIFF”. Chunk ini boleh terdiri daripada pelbagai jenis sub chunk yang lain. Subchunk yang mesti ada dalam sesebuah fail wave adalah subchunk format dan sub chunk data. Sub chunk format biasanya diletakkan sebelum sub chunk data.

2.3.1.5 Musical Instrument digital Interface (MIDI)

Fail jenis ini mempunyai extension “*.mid”. fail MIDI agak berbeza dengan fail jenis lain kerana ia merupakan multi platform dan hanya untuk muzik sahaja. Ianya bersifat fail instrumental dimana ia tidak mengandungi suara atau data audio yang sebenar. Ini menjadikan saiznya adalah kecil.[1,14,15,16,24,25,26,27]

Strukturanya hampir sama dengan struktur RIFF di mana ia juga membahagikan struktur fail ke dalam bentuk chunk. Terdapat 2 jenis chunk dalam struktur fail ini iaitu header chunks yang diwakili oleh perwakilan ASCII sebagai “MTrk”. Struktur bagi setiap chunk digambarkan dengan lebih terperinci dalam gambarajah di bawah.[20]

Chunk Pengepala (header Chunk)				
Jenis chunk	Panjang	Panjang = 6 byte		
4 byte (ASCII)	4 byte (32 bit binari)	16 bit	16 bit	16 bit
MThd	<length>	<format>	<tracks>	<division>

Rajah 2.7 : format Chunk Pengepala bagi fail MIDI

Term	Keterangan
Jenis chunk	<ul style="list-style-type: none"> - mengandung ID "MThd" - bersaiz 4 bait - menggunakan perwakilan ASCII
Panjang	<ul style="list-style-type: none"> - bersaiz 4 bait - perwakilan binary - mewakili saiz bagi chunk data
Data	<ul style="list-style-type: none"> - mengandungi 3 koleksi 16 bit data mengenai format, tracks dan division
<format>	<ul style="list-style-type: none"> - format bagi fail MIDI - terdapat 3 jenis format iaitu <ul style="list-style-type: none"> a) format = 0 sekiranya hanya ada satu single track. b) format = 1 sekiranya ada satu atau lebih track yang serentak di mana track pertama dikenali sebagai Tempo Map yang mengandungi semua jenis meta events

		c) format = 2 sekiranya ada satu
		atau lebih track boleh
		dimainkan secara
		berasingan

<tracks> - bilangan track chunk yang terdapat dalam fail

<division> - mengandungi pembahagian masa yang digunakan untuk menyahkodkan delta times bagi track event kepada masa nyata

- nilai ini diwakili oleh 2 format iaitu ticks per beat atau frames per second.

Jadual 2.3 : Penerangan mengenai struktur chunk
pengepala dalam fail MIDI

Track chunk		
Jenis	Panjang	Data
4 byte (ASCII)	4 byte (32 bit binari)	Panjang bytes (data binari)
MTrk	<length>	<delta time> <event>....

Rajah 2.8 : format chunk track bagi fail MIDI

Term	Keterangan
	mengandungi ID “MTrk”
Jenis	- bersaiz 4 bait - guna perwakilan ASCII
Panjang	- bersaiz 4 bait - mewakili saiz semua track iaitu bergantung kepada bilangan bait yang digunakan dalam semua event
Data	- boleh mengandungi 1 atau lebih pasangan <delta_time><event>
<delta_time>	- menentukan masa sesuatu event perlu dimainkan bergantung kepada event terakhir dalam track tersebut

-
- menentukan jujukan track dan bagaimana
<event> sesuatu track akan dimainkan
-

Jadual 2.4 : Penerangan mengenai struktur chunk Track

2.3.2 Perbandingan

Dalam membuat pemilihan format fail yang paling sesuai, perbandingan di buat dengan membandingkan beberapa ciri-ciri penting di antara setiap format fail yang berpotensi untuk digunakan. Terdapat tiga parameter utama untuk mendapatkan kualiti audio yang baik iaitu :

- Kadar persampelan (sampling rate)
- Bilangan bit per sampel
- Jenis channel yang digunakan.

Audio yang lebih berkualiti dikatakan mempunyai kadar persampelan yang tinggi, bilangan bit yang lebih besar dan juga menggunakan channel stereo.

<i>Nama fail</i>	<i>Saiz</i>	<i>Jenis data</i>	<i>Kualiti</i>	<i>Persekitaran penggunaan</i>	<i>Ciri lain</i>
Wave	besar	data yang tidak di padatkan	sangat berkualiti	- windows - sesuai untuk pemproses intel	

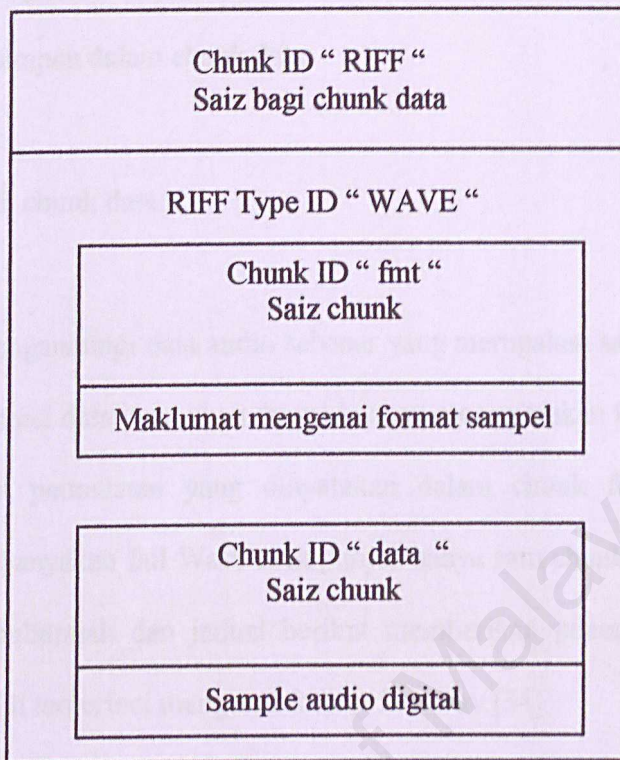
Windows Media/ASF	Lebih kecil	Pemadatan yang tinggi	Baik pada nada rendah	Windows	Guna pelbagai codec
MIDI	kecil	data yang tidak di padatkan	berkualiti	Multi platform	- Fail instrumental - perlu plug-in atau helper application untuk mainkan fail
MP3	Saiz kecil (2 kali ganda dari ASF)	Data yang dipadatkan	Kualiti baik (kualiti kurang jika pemadatan terlalu tinggi)	Windows	Hadkan bunyi kepada 16 MegaHz untuk buang kebisingan.
AIFF	Besar	Data yang tidak dipadatkan	Berkualiti	Macintosh dan Windows	

Jadual 2.5 : Perbandingan antara beberapa jenis format fail

2.3.3 Kesimpulan

Setelah membuat kajian dan perbandingan ke atas format fail yang ada, fail jenis wave dirasakan paling sesuai untuk digunakan dalam membangunkan projek ini. Fail jenis ini menyimpan data secara *least significant byte first*.

Secara amnya, binaan chunk header dan chunk track dalam fail wave digambarkan dalam rajah berikut[20,30]:



Rajah 2.9 : struktur chunk dalam fail Wave

Berdasarkan gambarajah di atas, chunk utama “RIFF” berjenis “WAVE” mempunyai 2 sub chunk yang lain iaitu sub chunk format yang diwakili oleh ID “fmt” dan sub chunk data yang diwakili oleh ID “data”.

a) Sub chunk format

Mengandungi maklumat mengenai bagaimana data disimpan dan bagaimana ia harus dimainkan. Ini

termasuklah jenis pemadatan yang digunakan, bilangan channel, kadar persampelan, bit per sampel dan sebagainya. Chunk ini menjelaskan format data yang disimpan dalam chunk data

b) Sub chunk data

Mengandungi data audio sebenar yang merupakan sampel-sampel data yang akan dinyahkodkan menggunakan format dan pemadatan yang dinyatakan dalam chunk format. Kebanyakan fail Wave mempunyai hanya satu chunk data. Gambarajah dan jadual berikut memberikan penerangan lebih terperinci mengenai format fail Wav.[34]

Bait offset		Saiz (bait)
0	ID Chunk	4
4	Saiz chunk	4
8	Format	4
12	ID bagi sub chunk	4
16	Saiz sub chunk	4
20	Format	2
22	Jenis channel	2
24	Kadar persampelan	4
28	Kadar bit	4
32	Block align	2
34	Kadar bit per sampel	2
36	ID bagi sub chunk 2	4
40	Saiz sub chunk 2	4
44	Data	Saiz sub chunk

Rajah 2.10 : Gambaran terperinci struktur chunk dalam fail

Wave

Term	Keterangan
ID Chunk	- ID " RIFF "
Saiz Chunk	- maklumat saiz fail - nilai ini tidak termasuk saiz bagi pengepala iaitu 8 bit untuk ID chunk dan juga saiz chunk.
Format	- tentukan jenis fail RIFF - contohnya " WAVE "
ID sub chunk	- ID bagi sub chunk terlibat - contohnya " fmt "
Saiz sub chunk	- biasanya 16 bait dan bait tambahan
Format	- tentukan jenis pemadatan yang digunakan berdasarkan kod tertentu.
Jenis channel	- Tentukan bilangan isyarat audio yang dikodkan kedalam chunk data - 1 sekiranya isyarat mono - 2 untuk isyarat stereo
Kadar persampelan	- bilangan sampel yang dimainkan dalam masa satu saat - tidak dipengaruhi oleh jenis channel
Kadar bit	- bilangan data Wave yang melalui D/A converter dalam masa satu saat untuk memainkan fail Wave
Blok align	- bilangan bait per sampel
Bit per sampel	- nilai untuk spesifikasikan bilangan bit yang digunakan untuk mencari setiap sampel - biasanya merupakan kumpulan 8 bit.
ID sub chunk 2	- ID bagi sub chunk - contohnya "data"
Saiz sub chunk 2	- bilangan bait data
Data	- mengandungi data audio sebenar

Jadual 2.6 : penerangan struktur chunk dalam fail Wave

2.3.4 Faktor Pemilihan

Terdapat beberapa faktor mengapa fail jenis Wave dirasakan paling sesuai untuk digunakan dalam projek ini. Ini termasuklah :

a) Kualiti yang baik

Walaupun fail ini bersaiz besar tetapi kualitinya sangat baik. Memandangkan saiz fail yang akan dibentuk dalam projek ini adalah kecil jadi faktor saiznya yang besar bukanlah satu penghalang untuk memilih menggunakan fail jenis ini.

b) Data yang tidak dipadatkan

Pemadatan yang dilakukan ke atas data biasanya sedikit sebanyak akan menjejaskan kualiti data sebenar. Dalam projek ini keaslian data adalah sangat penting bagi memastikan proses analisis mendapat hasil yang tepat.

c) Persekitaran teknologi yang sesuai

Fail ini bukan sahaja disokong oleh sistem pengoperasian yang bakal digunakan dalam projek ini malah juga sesuai untuk digunakan dalam persekitaran pemproses Intel.

- d) Menjadi pilihan kebanyakan pembangun projek dan sering kali digunakan untuk membangunkan projek yang berasaskan pengecaman suara (*voice recognition*)

KAJIAN PERALATAN ATAU TEKNIK YANG DIGUNAKAN.

2.4.1 Wavelet

Sebelum kemunculan Wavelet, analisa ke atas isyarat gelombang biasanya dilakukan dengan menggunakan Siri Fourier khususnya Penganalisa Fourier Masa Pendek (PFMP). PFMP ini membahagikan isyarat menggunakan tetingkap dan seterusnya mengira pekali fourier, Pekali fourier ini dipengaruhi oleh kedudukan tetingkap dan juga frekuensi yang ada pada pekali-pekali tersebut. Jika frekuensinya tinggi maka masa yang ada adalah singkat.

Kaedah PFMP ini mempunyai dua kelemahan utama iaitu[3, 7]:

- a) Memberikan maklumat frekuensi iaitu berapa banyak sesuatu frekuensi itu wujud dalam sesuatu isyarat tetapi tidak menyatakan apakah frekuensi yang wujud dalam masa tertentu.
- b) Kaedah ini hanya mampu menganalisa isyarat yang berkala sahaja.

[7]Wavelet adalah satu teknik atau kaedah yang muncul akibat daripada kelemahan Siri Fourier. Ia menggunakan kaedah pemotongan isyarat dengan membahagikan isyarat kepada beberapa bahagian kecil dan seterusnya menganalisa isyarat secara tidak bersandar.

Kaedah Wavelet yang menggunakan penganalisa wavelet ini secara ringkasnya dapat digambarkan melalui persamaan berikut:

$$\Phi_{(s,l)}(x) = 2^{-s/2} \Phi(2^{-s}x - l) \quad (1)$$

Fungsi ini merupakan fungsi rekursif di mana s dan l mewakili pekali untuk penskalaan dan translasi.

Untuk menguji pelbagai resolusi, penganalisa wavelet digunakan dalam fungsi penskalaan seperti di bawah:

$$W(x) = \sum_{k=-1}^{N-2} (-1)^k C_{k+1} \Phi(2x + k) \quad (2)$$

Di mana $W(x)$ merupakan fungsi penskalaan bagi penganalisa wavelet, Φ dan C_k adalah merupakan pekali wavelet. Ini mesti menepati syarat linear dan kuadratik iaitu:

$$W(x) = \sum_{k=0}^{N-2} C_k = 2 \quad (3)$$

$$W(x) = \sum_{k=0}^{N-2} C_k C_{k+2l} = 2\delta_{l,0} \quad (4)$$

2.3.1.1 Penukarganti Diskrit Wavelet

Penukarganti Diskrit Wavelet (PDW) adalah merupakan satu cara pemprosesan isyarat yang menggunakan kaedah penskalaan dan translasi yang dilakukan terhadap penganalisa wavelet.

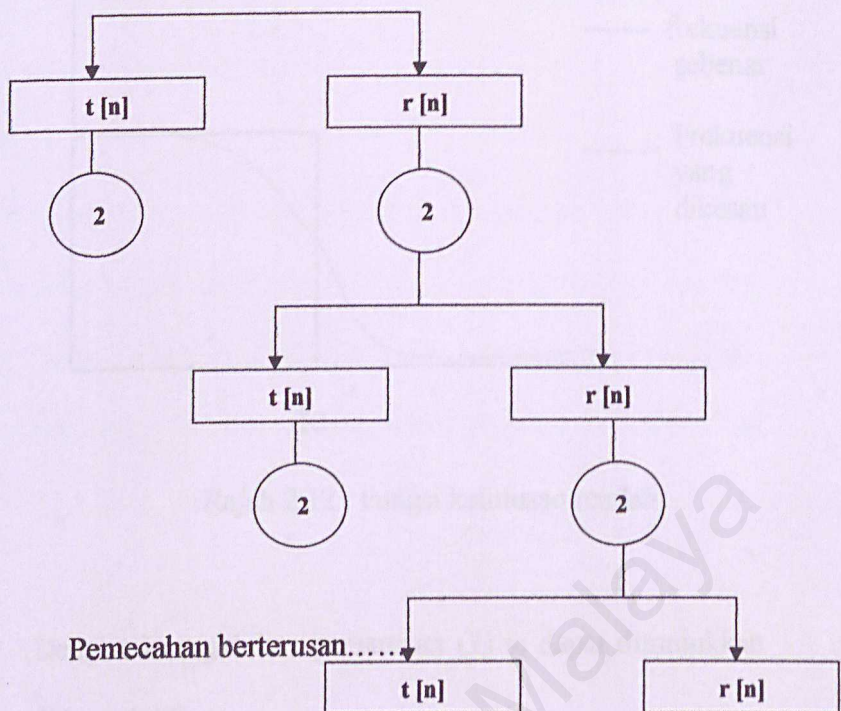
Di dalam konteks PDW penganalisa wavelet yang digunakan mestilah berbentuk diskrit seperti yang ditunjukkan di bawah:

$$\phi(x) = \sum_{k=0}^{M-1} C_k \phi(2x - k) \quad (5)$$

Di mana julat penambahan ditentukan oleh nilai positif M. Pekali c merupakan tertib wavelet tersebut. Syarat-syarat pekali adalah sama seperti yang ditunjukkan dalam persamaan (2) dan (3).

2.3.1.2 Algoritma Piramid

Algoritma ini hanya dapat diaplikasikan kepada saiz data N iaitu N adalah gandaan bagi dua. Data ini akan melalui 2 fungsi kelulusan rendah dan tinggi yang akan menghasilkan keluaran saiz yang setengah dari saiz asal. Algoritma ini dijelaskan melalui gambarajah di bawah [7]:



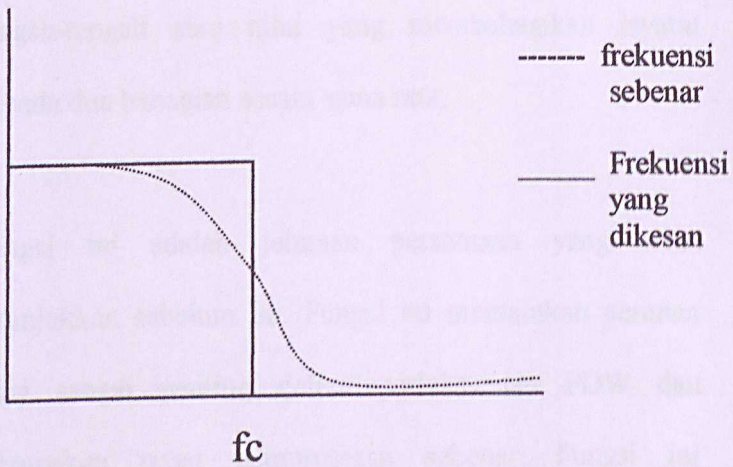
Rajah 2.11 : Algoritma pyramid

$t[n]$ adalah fungsi kelulusan peringkat tinggi manakala $r[n]$ adalah fungsi kelulusan peringkat rendah. Bahaian yang mengandungi frekuensi yang rendah mempunyai lebih banyak maklumat maka bahagian ini yang akan di proses berulang kali sehingga saiz keluaran menjadi tunggal.

2.3.1.3 Fungsi Kelulusan

Fungsi kelulusan tinggi dan rendah adalah seperti di bawah:

a) Fungsi kelulusan rendah

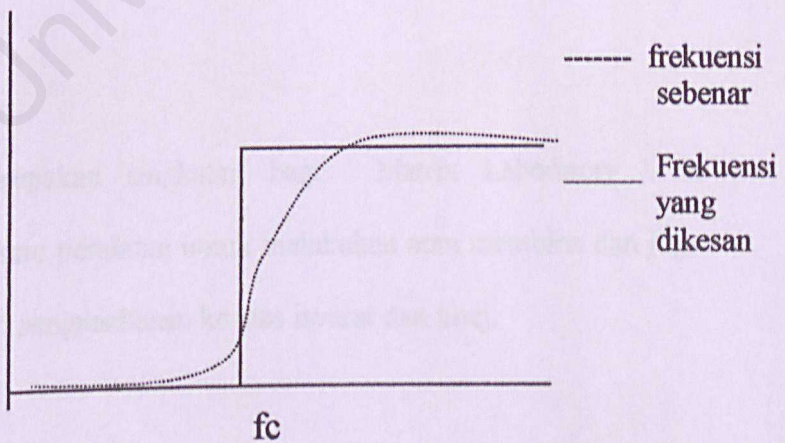


Rajah 2.12 : Fungsi kelulusan rendah

Dengan berpandukan persamaan (1) ia dapat ditunjukkan di bawah [7]:

$$a_i = \frac{1}{2} \sum_{j=1}^N C_{2i-j+1} f_1 \quad i=1.2.....N/2 \quad (6)$$

b) Fungsi kelulusan tinggi



Rajah 2.12 : Fungsi kelulusan tinggi

Fungsi ini mengesan nilai frekuensi yang melebihi nilai frekuensi pemotong (f_c). Nilai f_c merupakan nilai di tengah-tengah atau nilai yang membahagikan isyarat kepada dua bahagian secara sama rata.

Fungsi ini adalah jelmaan persamaan yang telah ditunjukkan sebelum ini. Fungsi ini memainkan peranan yang sangat penting dalam pelaksanaan PDW dan merupakan pusat pemprosesan sebenar. Fungsi ini menerima input input [7]:

Untuk menghasilkan keluaran berfrekuensi tinggi maka persamaan di bawah digunakan. Ianya dihasilkan berpandukan persamaan (2):

$$b_i = \frac{1}{2} \sum_{j=1}^N (-1)^{j+1} C_{2i-j+1} f_1 \quad i=1,2,\dots,N/2 \quad (7)$$

2.3.2 Matlab

Matlab merupakan singkatan bagi ' Matrix Laboratory '. Ia merupakan satu peralatan untuk melakukan atau membina dan juga menjalankan penganalisan ke atas isyarat dan imej.

Kaedah yang digunakan dalam penganalisa Siri Fourier. Ianya juga mampu mewakili hasil penganalisaan dengan baik dengan adanya fungsi untuk memplot graf tiga dimensi dan sebagainya [4, 5, 7,9].

2.3.3 Fast Fourier Transformation.

Sesuatu isyarat akan lebih mudah dikenalpasti atau di analisis. Sekiranya ia dapat dipecahkan kepada blok-blok yang lebih kecil. Algoritma Fast Fourier Transformation (FFT) ini merupakan salah satu kaedah untuk tujuan tersebut.[4]

FFT menganalisis isyarat dan menghasilkan perwakilan amplitud dan frekuensi bagi isyarat tersebut.

2.4 RINGKASAN

Secara keseluruhannya bab ini memaparkan gambaran menyeluruh mengenai projek yang bakal dibangunkan, Ini termasuklah segala kajian serta penyelidikan awal bagi mengumpul segala maklumat dan data-data yang diperlukan.

Penyelidikan atau kajian yang dilakukan dipecahkan kepada tiga bahagian utama iaitu penyelidikan ke atas suara manusia, kajian format fail dan juga penyelidikan untuk mendapatkan peralatan atau teknik yang bersesuaian yang boleh digunakan untuk membangunkan projek. Data - data yang diperolehi di bandingkan untuk mendapatkan pilihan yang paling tepat.

Di akhir bab ini beberapa keputusan dibuat iaitu :

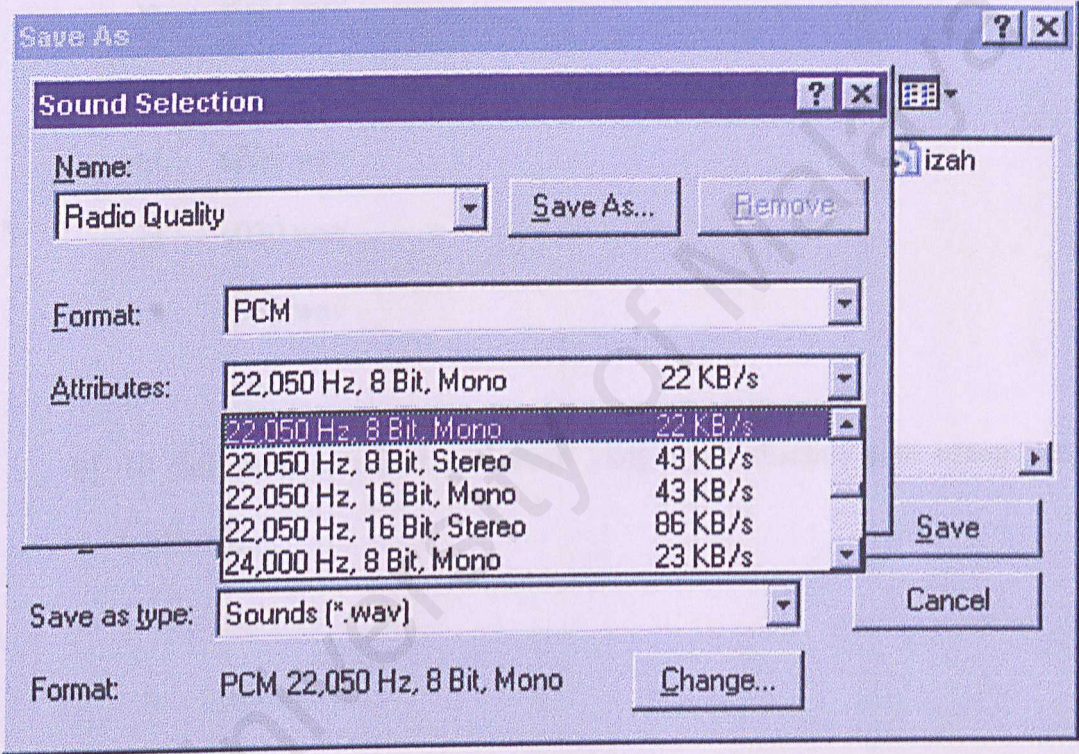
- a) Format fail yang digunakan adalah fail Wave
- b) Teknologi pengecaman suara adalah berkonsepkan Speaker Dependent dan juga Text Independent.
- c) Peralatan utama yang akan digunakan termasuklah Wavelet, Matlab juga Algoritma Fast Fourier Transformation.

Bab 3

Perlaksanaan Rekabentuk

3.1 PEMBENTUKAN FAIL DATA

Untuk membentuk fail data yang diperlukan, suara dirakamkan dengan menggunakan perakam suara MP3. Suara yang dirakamkan perlu ditukar ke dalam format PCM dengan kadar persampelan sebanyak 22,050 Hz dan channel adalah mono. Ini dilakukan dengan menyimpan rekod suara tersebut sekali lagi (*resave*) seperti yang ditunjukkan dalam rajah di bawah:



Rajah 3.1 : Penukaran fail dari perakam suara MP3 kepada bentuk wav

Terdapat 30 sampel suara yang direkodkan untuk digunakan sebagai fail data dalam projek ini. Fail-fail tersebut adalah seperti berikut :

a) 10 fail yang merekod bagi perkataan yang sama oleh orang yang sama

iaitu :

- i. v022.wav
- ii. v023.wav
- iii. v024.wav
- iv. v025.wav
- v. v026.wav
- vi. v027.wav
- vii. v028.wav
- viii. v029.wav
- ix. v030.wav
- x. v031.wav

b) 10 fail yang merekod perkataan yang sama disebut oleh orang yang berlainan

- i. jua.wav
- ii. jaja.wav
- iii. as.wav
- iv. farin.wav
- v. hanan.wav
- vi. izah.wav
- vii. naza.wav
- viii. tini.wav

ix. ana.wav

x. hajar.wav

c) 10 fail yang merekod perkataan yang berlainan disebut oleh orang yang sama

i. open.wav

ii. rejected.wav

iii. enter system.wav

iv. accepted.wav

v. hello.wav

vi. buka.wav

vii. tutup.wav

viii. suara.wav

ix. masuk.wav

x. bunyi.wav

3.2 MEMBACA FAIL DATA

Fail-fail yang di bentuk perlu dibaca untuk mendapatkan data-data yang diperlukan. Fail-fail ini dibaca dengan menggunakan aturcara yang di tulis melalui Matlab. Maklumat-maklumat yang diperlukan dari fail-fail tersebut termasuklah :

Maklumat asas dari *header fail*

Maklumat-maklumat yang diperolehi serta penerangannya dicatatkan dalam jadual berikut :

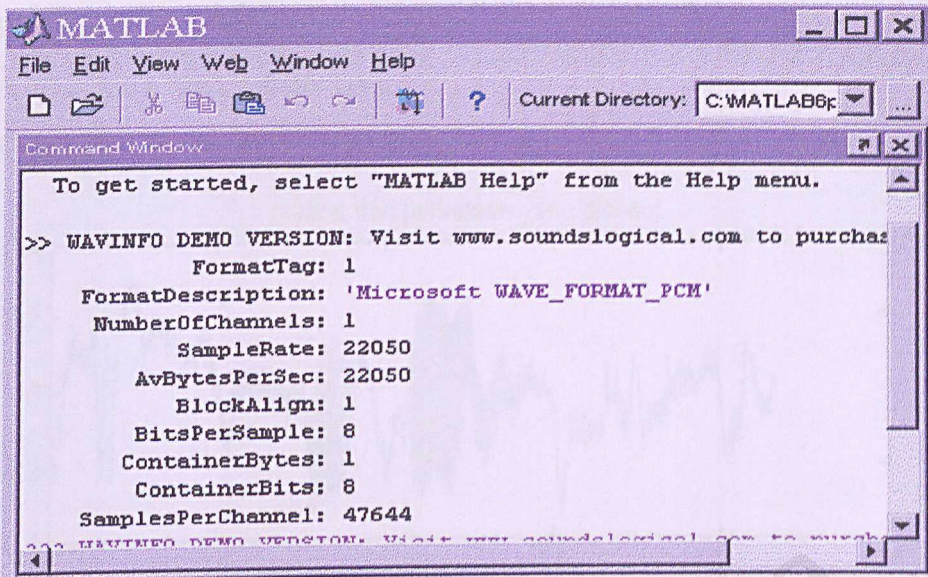
Data	Penerangan
<i>FormatTag</i>	<ul style="list-style-type: none">• Tag atau nilai yang mewakili jenis format tertentu . Dalam kes ini <i>FormatTag</i> bagi semua mempunyai nilai 1
<i>FormatDescription</i>	<ul style="list-style-type: none">• Jenis format data yang lebih spesifik• Dalam kes ini kesemua data adalah jenis PCM iaitu jenis data sebenarnya yang tidak mengalami <i>compression</i>
<i>NumberOfChannels</i>	<ul style="list-style-type: none">▪ Bilangan channel▪ Kesemua data mempunyai hanya satu channel (mono)

<i>SampleRate</i>	<ul style="list-style-type: none"> ▪ Bilangan sampel dalam masa satu saat ▪ Kesemua data mempunyai kadar persampelan sebanyak 22050 Hz yang bermaksud sebanyak 22050 sampel dalam sesaat.
<i>AvBytesPerSec</i>	<ul style="list-style-type: none"> ▪ Purata bytes dalam masa satu saat ▪ $\text{SampleRate} * \text{BlockAlign}$ ▪ 22050 bagi semua data
<i>BlockAlign</i>	<ul style="list-style-type: none"> ▪ bilangan bytes dalam satu sampel ▪ semua fail data mempunyai nilai 1
<i>BitsPerSample</i>	<ul style="list-style-type: none"> ▪ bilangan bit dalam satu sampel ▪ semua fail data mengandungi 8 bit dalam satu sampel
<i>SampelsPerChannel</i>	<ul style="list-style-type: none"> ▪ bilangan sampel bagi satu channel ▪ mempunyai nilai yang berbeza bagi setiap fail data

Jadual 3.1 : Maklumat dari header fail data.

Berikut adalah aturcara yang ditulis serta contoh paparan keluaran atau maklumat yang diperolehi dari fail open.wav :

```
a = wavinfo ('C:\Documents and Settings\UM\My Documents\aslizawani\
fail wav \open.wav');
```

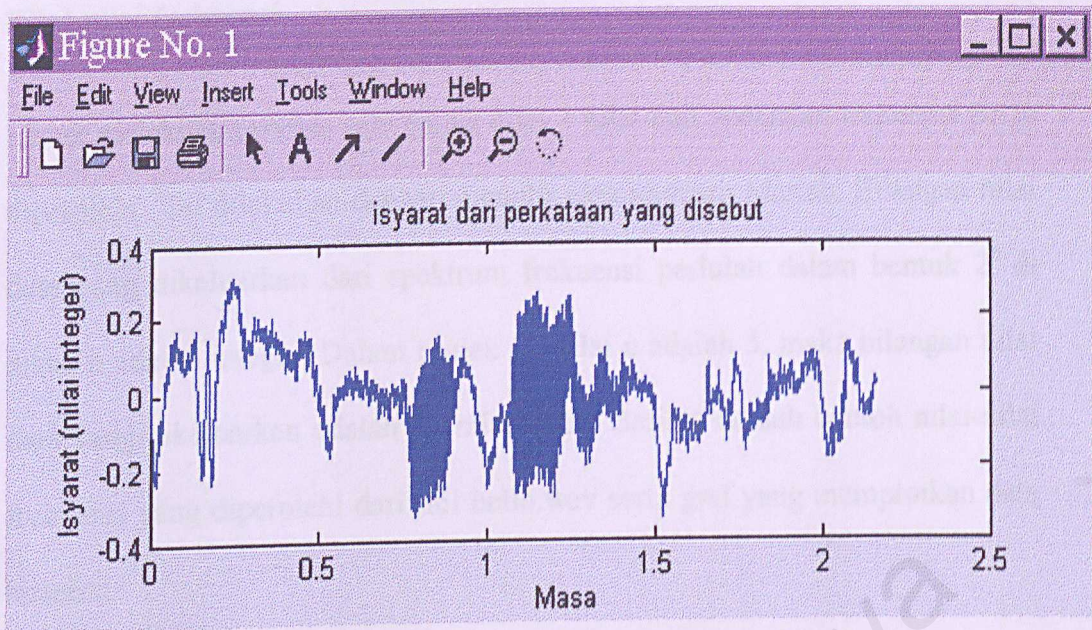
Rajah 3.2 : Paparan output dari *header* fail

Maklumat yang diperolehi dari fail-fail yang lain boleh didapati daripada lampiran B.

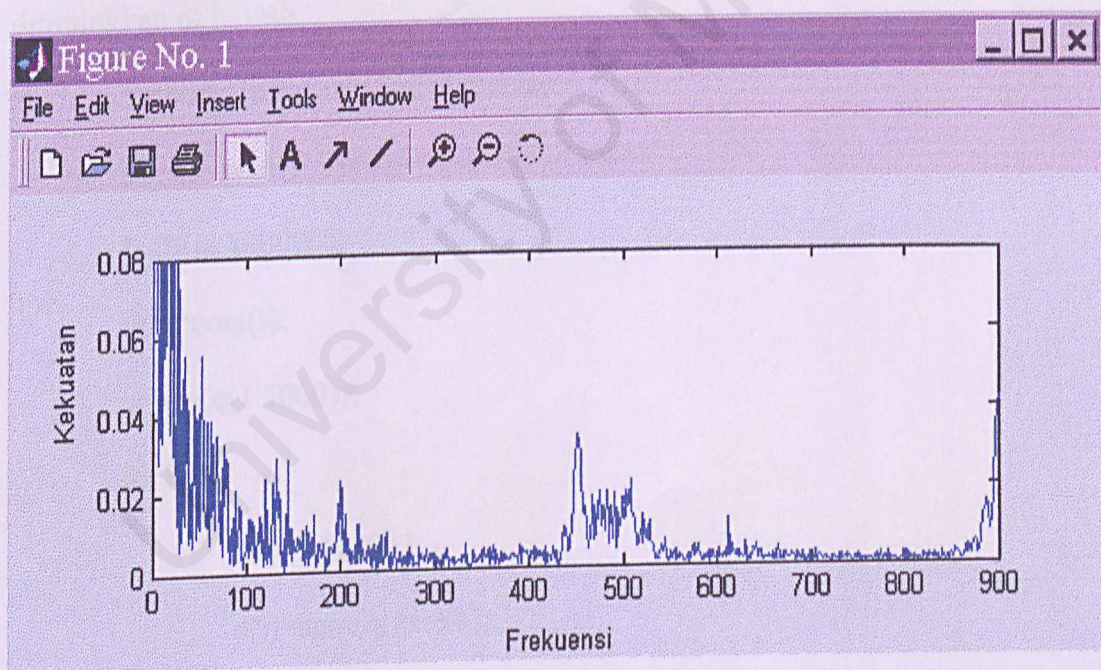
Spektrum frekuensi

Spektrum frekuensi merupakan gambarajah pemetaan isyarat daripada suara yang direkodkan dalam bentuk graf. Ia diperolehi dengan menggunakan *Fast Fourier Transform*. Satu aturcara matlab dibentuk yang akan memanggil fungsi *fft* yang mewakili *Fast Fourier Transform* [8.9].

Contoh spektrum frekuensi yang diperolehi daripada fail 'open.wav' adalah seperti yang ditunjukkan dalam gambarajah di bawah.



Rajah 3.3 : Isyarat yang terbentuk daripada suara yang telah direkodkan



Rajah 3.4 : Spektrum frekuensi yang terbentuk

Aturcara yang terlibat dan juga contoh-contoh spektrum bagi fail lain adalah seperti dalam lampiran C.

Nilai-nilai frekuensi

Untuk melakukan proses seterusnya nilai – nilai dari spektrum frekuensi perlu diperoleh. Ini dilakukan dengan menulis satu aturcara Matlab. Bilangan nilai data yang dikeluarkan dari spektrum frekuensi perlulah dalam bentuk 2^n di mana n adalah integer. Dalam projek ini nilai n adalah 5, maka bilangan nilai data yang dikeluarkan adalah 32 nilai data. Berikut adalah contoh nilai-nilai frekuensi yang diperolehi dari fail hello.wav serta graf yang memplotkan data tersebut.

Aturcara yang digunakan serta output yang dipaparkan adalah seperti yang ditunjukkan di bawah :

```
function [p]=soundSig(filename)
in=wavread('F:\As\tesis\data2\izah.wav');
f=fft(in,100000);
q=f.*conj(f);
q=abs(q(1:5000));
for i=1:4
    t=(i-1)*100+1;
    p(i)=sum(q(t:t+99));
end
p=p/sum(p);
plot(p)
```


Data –data yang diperolehi adalah seperti berikut :

Columns 1 through 7

0.1140 0.0349 0.0652 0.0948 0.0221 0.0117 0.0012

Columns 8 through 14

0.0020 0.2386 0.1795 0.0824 0.0094 0.0022 0.0013

Columns 15 through 21

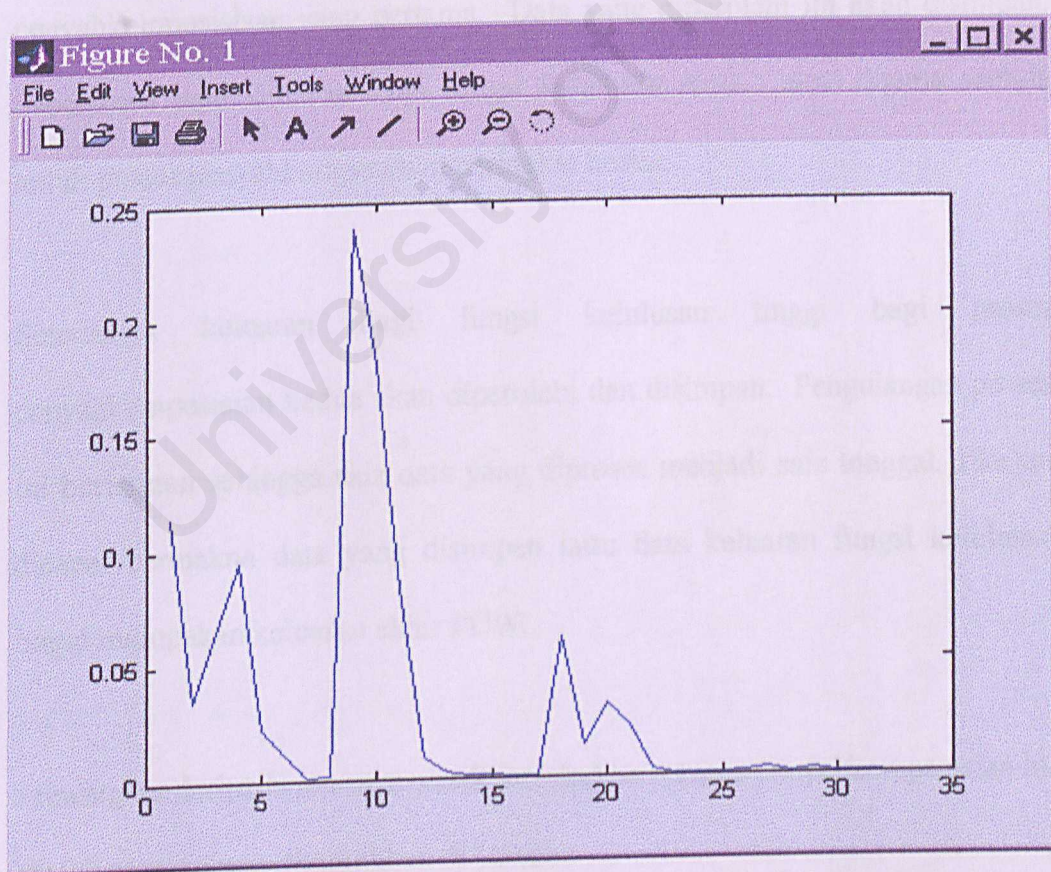
0.0010 0.0006 0.0006 0.0601 0.0134 0.0313 0.0217

Columns 22 through 28

0.0028 0.0006 0.0007 0.0002 0.0013 0.0026 0.0006

Columns 29 through 32

0.0020 0.0006 0.0004 0.0001



Rajah 3.5 : graf bagi nilai spektrum frekuensi

3.3 PROSES PENYAHKOMPOSISIAN (*DECOMPOSITION*)

Proses ini akan menyahkomposisikan isyarat mentah seterusnya menghasilkan keluaran akhir PDW. Ianya dijalankan menggunakan wavelet [7,8].

Pada peringkat penyahkomposisian pertama atau peringkat kelulusan pertama, data yang ada akan diproses oleh fungsi kelulusan rendah iaitu *lowpass* dan fungsi kelulusan tinggi iaitu *highpass*. Kedua-dua fungsi ini mempunyai nilai keluaran masing-masing.

Keluaran dari fungsi kelulusan tinggi merupakan hasil keluaran penyahkomposisian yang pertama. Data yang diperolehi ini akan disimpan. Sebaliknya data keluaran bagi fungsi kelulusan rendah akan diguna semula untuk proses penyahkomposisian peringkat kedua.

Seterusnya keluaran bagi fungsi kelulusan tinggi bagi proses penyahkomposisian kedua akan diperolehi dan disimpan. Pengulangan proses ini berlarutan sehingga saiz data yang diproses menjadi saiz tunggal. Jika ini dicapai bermakna data yang disimpan iaitu data keluaran fungsi kelulusan tinggi merupakan keluaran akhir PDW.

Sebahagian daripada aturcara untuk melakukan peroses penyahkomposisian ini adalah seperti yang ditunjukkan di bawah :


```

s = sin (2*pi*v1*t) + (2*pi*v2*t) + (2*pi*v3*t);

g0 = [ 0.9505; 0.0351; 0.0111; 0.0032];

k = [0; 1; 2; 3];

% penapis (filters) untuk proses penyahkomposisian dan pembinaan

% semula

g1 = flipud (g0).*(-1).^k;

h0 = flipud (g0) / 2;

h1 = flipud (g1) / 2;

% penyahkomposisian pertama

x = conv (s,h0);

a0 = x(1:2:length(x));

x = conv (s,h1);

w0 = x(1:2:length (x));

% penyahkomposisian kedua

x = conv (a0,h0);

a1 = x(1:2:length (x));

x = conv (a0,h1);

w1 = x(1:2:length (x));

```

Dalam aturcara di atas, s merupakan nilai bagi isyarat sebenar atau isyarat mentah yang akan dinyahkomposisikan dan di bina semula. Nilai g0 pula adalah nilai yang diperolehi daripada spektrum frekuensi. *Flipud* merupakan

penapis (*filters*) yang digunakan dalam proses penyahkomposisian dan pembinaan semula.

Setelah tamat penyahkomposisian peringkat pertama , proses peringkat kedua diteruskan dengan panjang isyarat sebenar (s) telah menjadi separuh dari saiz asal ($a0$).

3.4 PEMBINAAN SEMULA (*RECONSTRUCTION*)

Proses pembinaan semula merupakan proses untuk membina semula isyarat mentah yang telah dinyahkomposisikan. Nilai data yang diperolehi daripada proses ini sepatutnya sama dengan nilai data isyarat mentah sebelum dinyahkomposisikan.

Sebahagian daripada aturcara untuk melakukan proses pembinaan semula ini adalah seperti yang ditunjukkan di bawah. Nilai *a0_rec* merupakan nilai pembinaan semula untuk *a0* manakala nilai *s_rec* merupakan nilai binaan semula bagi isyarat mentah, *s*.

```
% pembinaan semula peringkat kedua
x = zeros (2*length (a1),1);
x(1:2:2*length (a1)) = a1(1: length (a1));
y = zeros (2*length (w1),1);
y(1:2:2*length (w1)) = w1 (1: length (w1));
x = conv (x,g0) + conv (y,g1);
a0_rec = x (4:length(x)-4);

% pembinaan semula peringkat pertama
y = zeros (2*length (w0),1);
y(1:2:2*length (w0)) = w0 (1: length (w0));
x = zeros (2*length (a0_rec),1);
```

```
x(1:2:2*length(a0_rec)) = a0_rec;
```

```
x = conv(x,g0)
```

```
y = conv(y,g1);
```

```
y = x(1:length(y)) + y;
```

```
s_rec = y(4:length(y)-4);
```

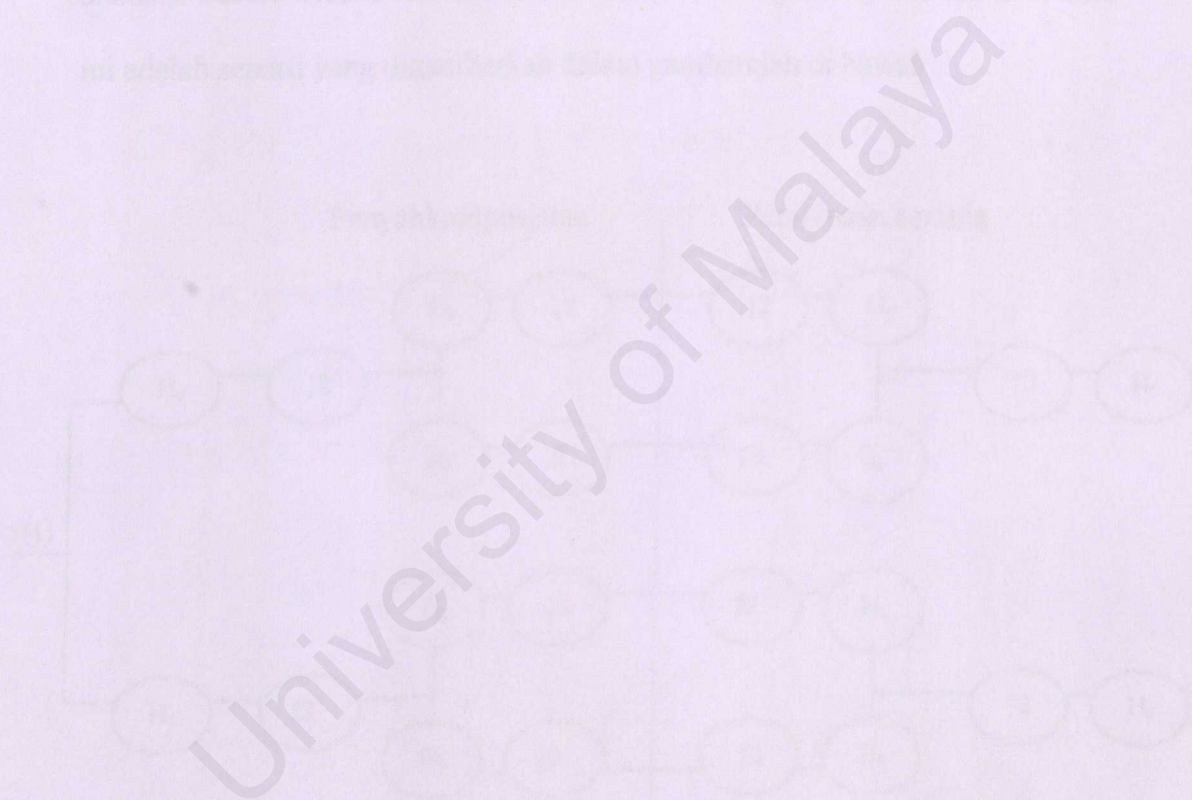
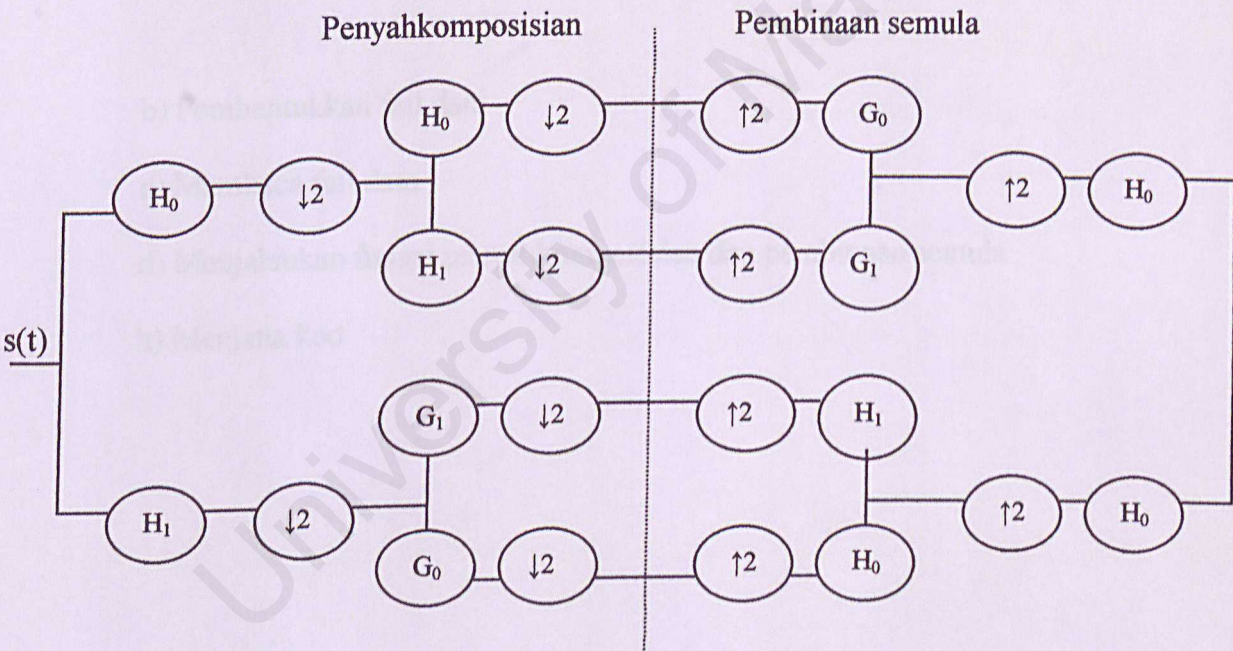


Figure 3.1: Diagram showing the sequence of nodes in the network.

3.5 PENGHASILAN NILAI UNIK

Nilai unik x merupakan paparan output akhir bagi aturcara yang menjalankan fungsi penyahkomposisian dan pembinaan semula. Keseluruhan aturcara ini boleh dilihat di bahagian lampiran D.

Penghasilan kod unik ini akan disentuh dengan lebih terperinci di bahagian analisis. Secara keseluruhannya kedua-dua proses bagi menghasilkan nilai unik ini adalah seperti yang digambarkan dalam gambarajah di bawah :



Rajah 3.6 : proses penyahkomposisian dan pembinaan semula dalam wavelet.

3.6 RINGKASAN

Secara keseluruhannya bab ini memberi gambaran bagaimana projek ini dibangunkan atau dengan kata lain bab ini cuba menerangkan satu persatu langkah-langkah pembangunan projek. Ia juga secara tidak langsung menghuraikan bagaimana peralatan yang telah ditentukan dalam bab sebelum ini digunakan dalam aktiviti membangunkan projek.

Benkut disenaraikan langkah-langkah yang terlibat dalam aktiviti pembangunan projek yang telah diuraikan dalam bab ini:

- b) Pembentukan fail data
- c) Membaca fail data
- d) Menjalankan fungsi penyahkomposisian dan pembinaan semula
- h) Menjana kod

Salah satu hal yang harus diperhatikan dalam proses belajar mengajar adalah bagaimana meningkatkan motivasi siswa. Salah satu cara untuk meningkatkan motivasi siswa adalah dengan memberikan tugas yang menantang. Dengan memberikan tugas yang menantang, siswa akan merasa bahwa tugas tersebut penting dan mereka akan berusaha untuk menyelesaikannya. Selain itu, memberikan tugas yang menantang juga dapat membantu siswa untuk mengembangkan keterampilan pemecahan masalah mereka. Dengan memberikan tugas yang menantang, siswa akan belajar bagaimana memecahkan masalah yang sulit dan mereka akan merasa bangga dengan pencapaian mereka.

Bab 4

Analisis

4.1 PENGENALAN

Dalam bab ini, analisis akan dilakukan ke atas nilai unik (x) bagi setiap sampel data untuk melihat perkaitan atau menilai nilai unik ini melalui perbezaan yang dihasilkan dalam setiap sampel. Dengan erti kata lain, perbezaan dikaji antara sampel yang disebut oleh orang yang sama menggunakan perkataan yang sama, orang yang berlainan menggunakan perkataan yang sama dan orang yang sama menggunakan perkataan yang berlainan.

Apa yang ingin diperolehi daripada analisis ini adalah untuk mengenal pasti nilai data yang unik bagi individu tertentu yang boleh membezakannya dari individu yang lain. Nilai unik ini boleh digunakan dalam banyak aplikasi terutamanya dalam bidang keselamatan sistem di mana nilai ini boleh menjadi pengecam bagi individu tersebut.

Ini berlaku apabila individu menyebut sesuatu perkataan yang akan diproses dan dibandingkan dengan nilai unik ini. sekiranya nilai yang terhasil sama dengan nilai unik ini maka identiti individu tersebut disahkan.

4.2 ANALISIS KE ATAS NILAI UNIK (x)

Secara amnya analisis yang dilakukan mengambil kira semua sampel yang secara keseluruhannya dibahagikan kepada 3 kumpulan sampel utama seperti berikut :

1. sampel 1 yang mengandungi 10 sampel rekod perkataan yang sama disebut oleh orang yang sama.
2. sampel 2 yang mengandungi 10 sampel rekod perkataan yang sama yang disebut oleh orang yang berlainan.
3. sampel 3 yang mengandungi 10 sampel rekod perkataan yang berlainan yang disebut oleh orang yang sama.

Dalam analisis ini , satu fail rekod yang menyebut perkataan 'open' dijadikan sebagai fail ujian untuk membuat perbandingan dengan fail fail yang lain. Ini bertujuan untuk mendapatkan kiraan ralat yang wujud dalam setiap nilai data yang diperolehi. Analisis ini juga mengambil kira perbezaan nilai unik (x) di antara setiap sampel fail data dalam setiap sampel 1, sampel 2 dan sampel 3 serta memaparkan perbezaan yang lebih ketara antara ketiga-tiga kumpulan sampel yang lebih besar.

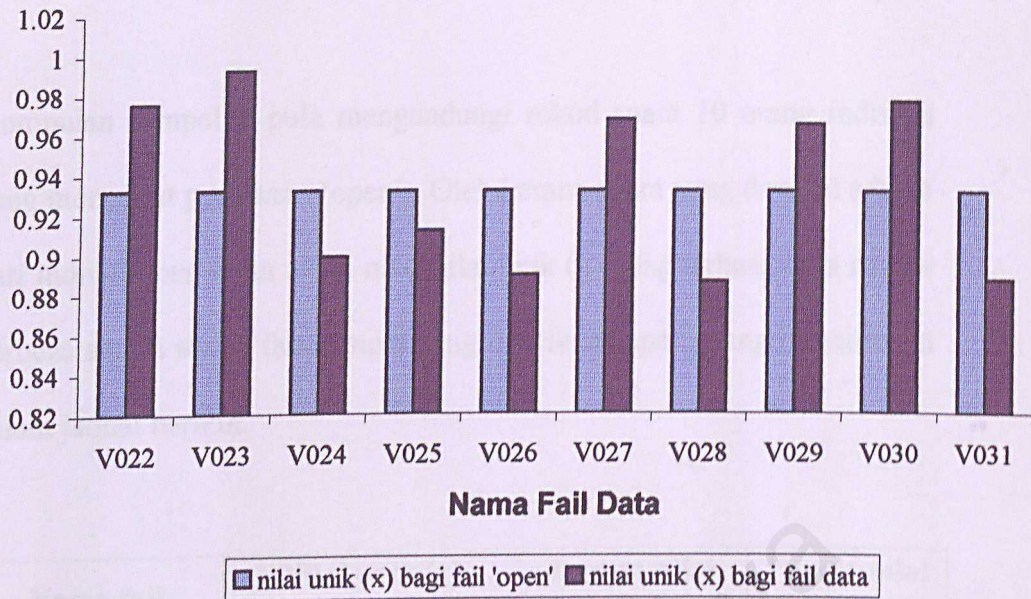
4.2.1 Kumpulan sampel 1

Sampel data dalam kumpulan ini adalah rekod suara yang menyebut perkataan “open” oleh seorang individu sahaja. Nilai unik (x) yang dihasilkan oleh setiap sampel sepatutnya mempunyai nilai yang sama antara satu sampel dengan sampel yang lain.

Walau bagaimanapun nilai yang terhasil menunjukkan wujudnya ralat dalam nilai data. Namun begitu memandangkan ianya adalah rekod suara individu yang sama, ralat yang wujud tidaklah begitu besar iaitu antara 0.01 hingga 0.05. Jadual dibawah memaparkan purata nilai unik (x) yang terhasil bagi setiap sampel dalam kumpulan sampel ini serta ralat bagi setiap fail data apabila dibandingkan dengan fail ujian ‘open’.

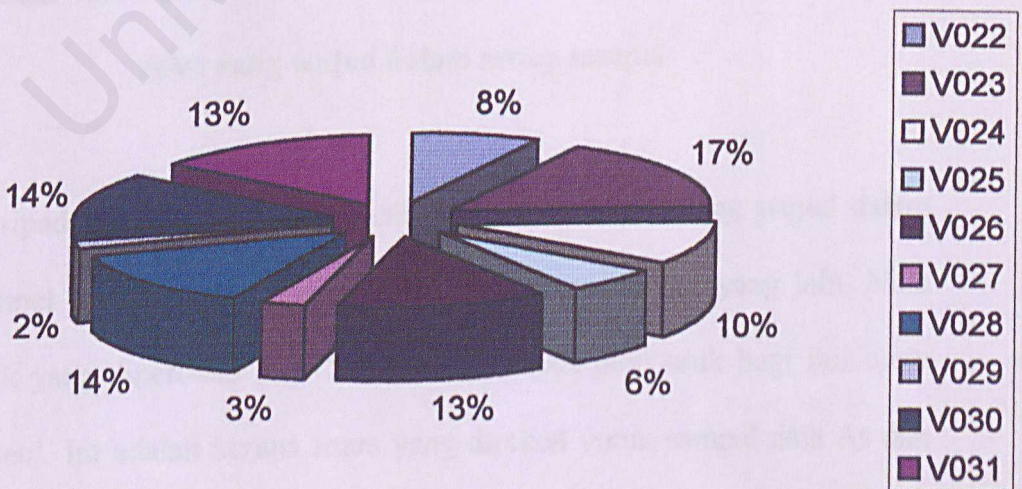
Nama fail	Nilai unik (x) Bagi fail ‘open’	Purata nilai unik (x)	Purata nilai ralat
V022	0.9325	0.9758	0.0265
V023	0.9325	0.9929	0.0564
V024	0.9325	0.8996	0.0330
V025	0.9325	0.9128	0.0198
V026	0.9325	0.8900	0.0426
V027	0.9325	0.9682	0.0093
V028	0.9325	0.8868	0.0458
V029	0.9325	0.9663	0.0061
V030	0.9325	0.9781	0.0457
V031	0.9325	0.8881	0.0445
Purata nilai ralat untuk 10 sampel			0.0329

Jadual 4.1: Purata nilai unik bagi setiap sampel kumpulan 1 serta ralat yang wujud



Rajah 4.1 : Perbandingan purata nilai (x) bagi setiap sampel dalam kumpulan 1 dengan fail 'open'

Graf di atas menunjukkan dengan lebih jelas kewujudan ralat dalam nilai-nilai unik (x) bagi setiap sampel dalam kumpulan ini. Carta pai di bawah pula menunjukkan peratusan ralat yang wujud dalam setiap sampel.



Rajah 4.2 : Peratusan Ralat yang wujud dalam setiap sampel kumpulan 1

4.2.2 Kumpulan sampel 2

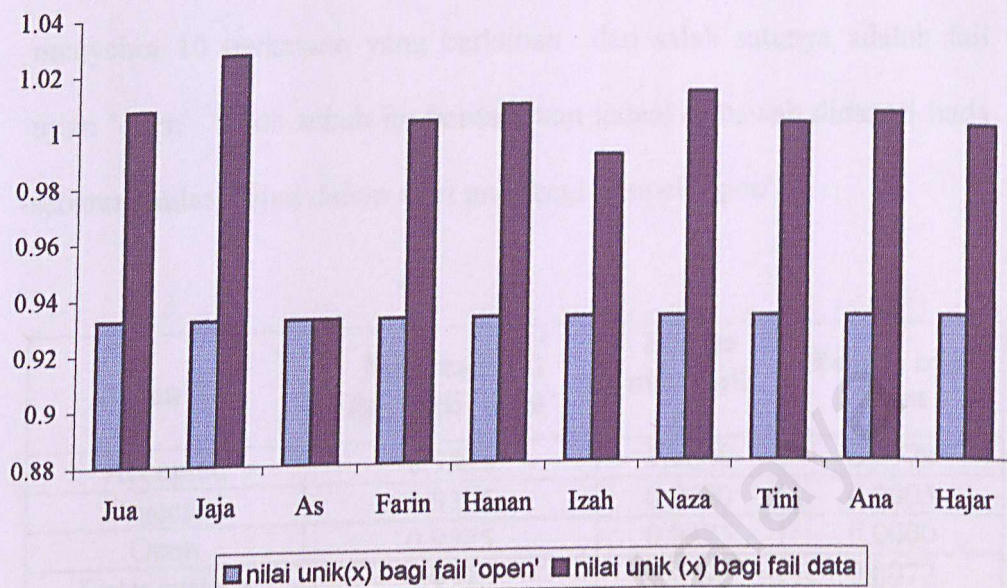
Kumpulan sampel 2 pula mengandungi rekod suara 10 orang individu yang menyebut perkataan ‘open’. Oleh kerana suara yang direkod adalah dari individu berlainan maka nilai-nilai unik (x) yang terhasil juga adalah berbeza antara setiap fail sampel yang dilarikan seperti yang ditunjukkan dalam jadual berikut.

Nama fail	Nilai unik (x) Bagi fail ‘open’	Purata nilai unik (x)	Purata nilai ralat
Jua	0.9325	1.0072	0.0846
Jaja	0.9325	1.0273	0.1235
As	0.9325	0.9910	0.0586
Farin	0.9325	1.0030	0.0760
Hanan	0.9325	1.0086	0.0868
Izah	0.9325	1.0074	0.0750
Naza	0.9325	1.0130	0.0953
Tini	0.9325	1.0018	0.0710
Ana	0.9325	1.0056	0.0805
Hajar	0.9325	1.0006	0.0735
Purata nilai ralat untuk 10 sampel			0.0825

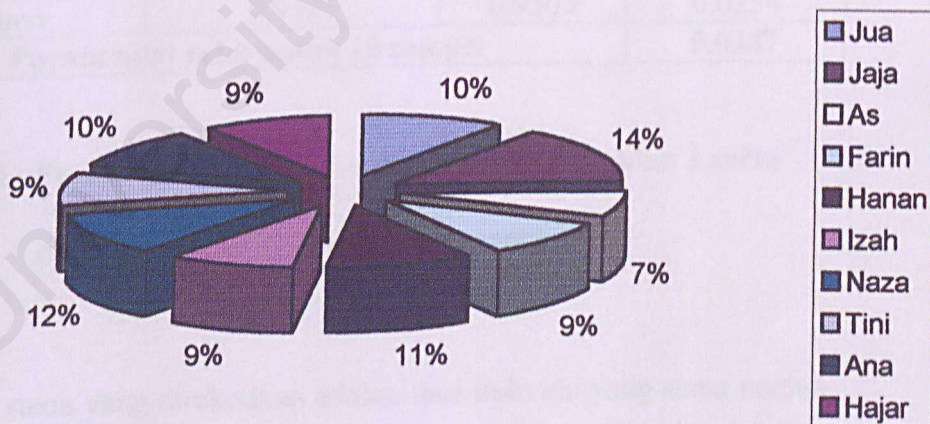
Jadual 4.2 : Purata nilai unik bagi setiap sampel kumpulan 2 serta ralat yang wujud dalam setiap sampel

Daripada jadual di atas, didapati bahawa nilai ralat yang wujud dalam sampel data As adalah lebih kecil berbanding dengan yang lain. Nilai unik yang diperolehi juga lebih menghampiri nilai unik bagi fail ujian ‘open’. Ini adalah kerana suara yang direkod untuk sampel data As dan juga sampel ujian ‘open’ adalah suara dari orang yang sama.

Keadaan ini kelihatan lebih ketara melalui graf serta carta pai yang ditunjukkan berikut.



Rajah 4.3 : Perbandingan purata nilai (x) bagi setiap sampel dalam kumpulan 2 dengan fail 'open'



Rajah 4.4 : Peratusan Ralat yang wujud dalam setiap sampel kumpulan dua.

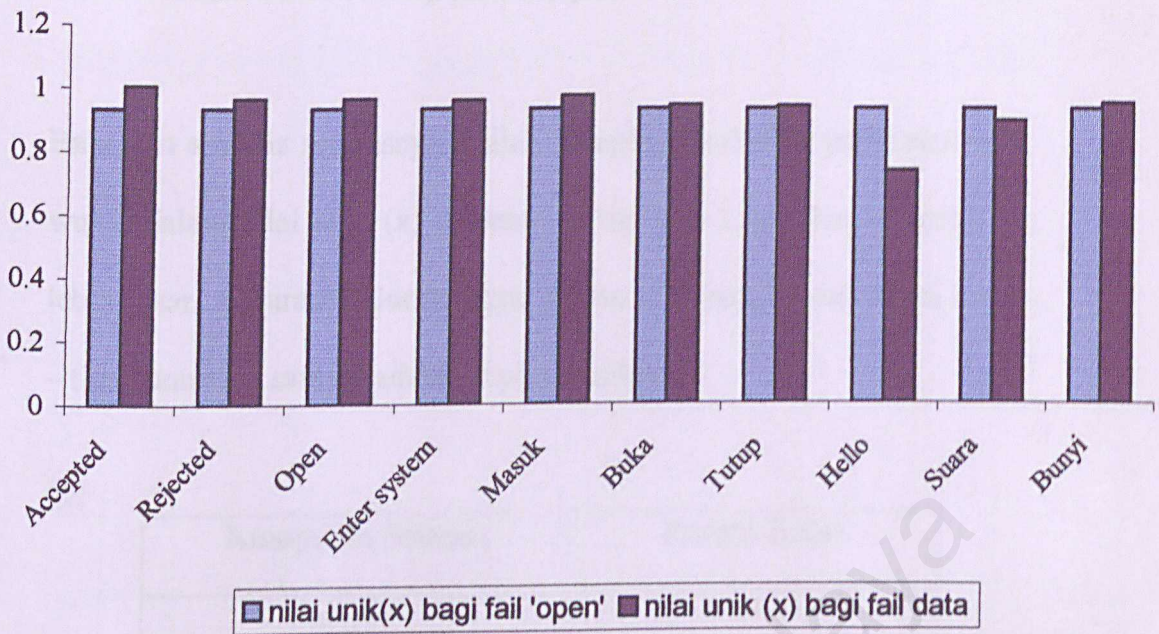
4.2.3 Kumpulan sampel 3

Kumpulan sampel 3 pula merekod suara dari individu yang sama yang menyebut 10 perkataan yang berlainan dan salah satunya adalah fail ujian ‘open’. Oleh sebab itu berdasarkan jadual di bawah didapati tiada sebarang ralat wujud dalam nilai unik bagi sampel ‘open’.

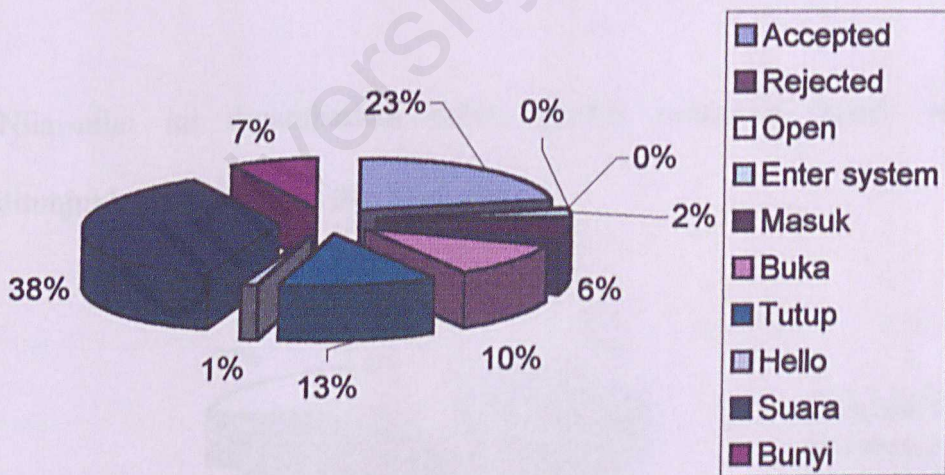
Nama fail	Nilai unik (x) Bagi fail ‘open’	Purata nilai unik (x)	Purata nilai ralat
Accepted	0.9325	1.0040	0.0779
Rejected	0.9325	0.9630	0.0005
Open	0.9325	0.9325	0.0000
Enter system	0.9325	0.9596	0.0072
Masuk	0.9325	0.9751	0.0218
Buka	0.9325	0.9447	0.0363
Tutup	0.9325	0.9398	0.0458
Hello	0.9325	0.9373	0.0048
Suara	0.9325	0.8956	0.1282
Bunyi	0.9325	0.9503	0.0254
Purata nilai ralat untuk 10 sampel			0.0347

Jadual 4.3 : Purata nilai unik bagi setiap sampel kumpulan 3 serta ralat yang wujud dalam setiap sampel

Walaupun suara yang direkodkan adalah dari individu yang sama namun nilai unik yang terhasil masih menunjukkan perbezaan yang lebih tinggi berbanding dengan sampel dalam kumpulan satu. Perbezaan ini ditunjukkan dalam graf seperti di bawah manakala peratusan ralat yang wujud digambarkan menggunakan carta pai berikut:



Rajah 4.5 : Perbandingan purata nilai (x) bagi setiap sampel dalam kumpulan 3 dengan fail 'open'



Rajah 4.6 : Peratusan Ralat yang wujud dalam setiap sampel kumpulan tiga.

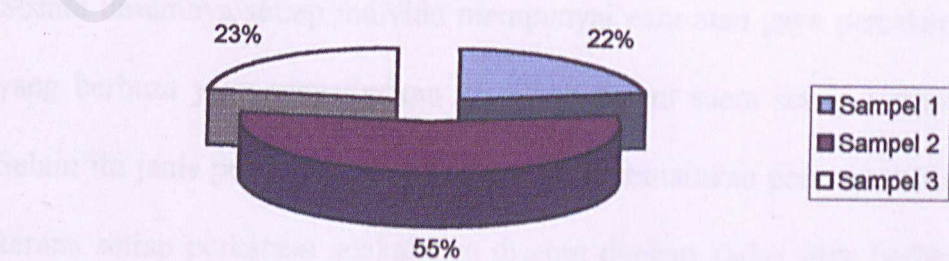
4.2.4 Perbandingan antara kumpulan sampel

Bahagian analisis seterusnya adalah mengkaji perkaitan perbezaan yang wujud dalam nilai unik (x) di antara ketiga-tiga kumpulan sampel yang lebih besar. Secara keseluruhannya purata ralat yang wujud dalam ketiga – tiga kumpulan sampel adalah seperti berikut :

Kumpulan Sampel	Purata Ralat
Sampel 1	0.0329
Sampel 2	0.0825
Sampel 3	0.0347

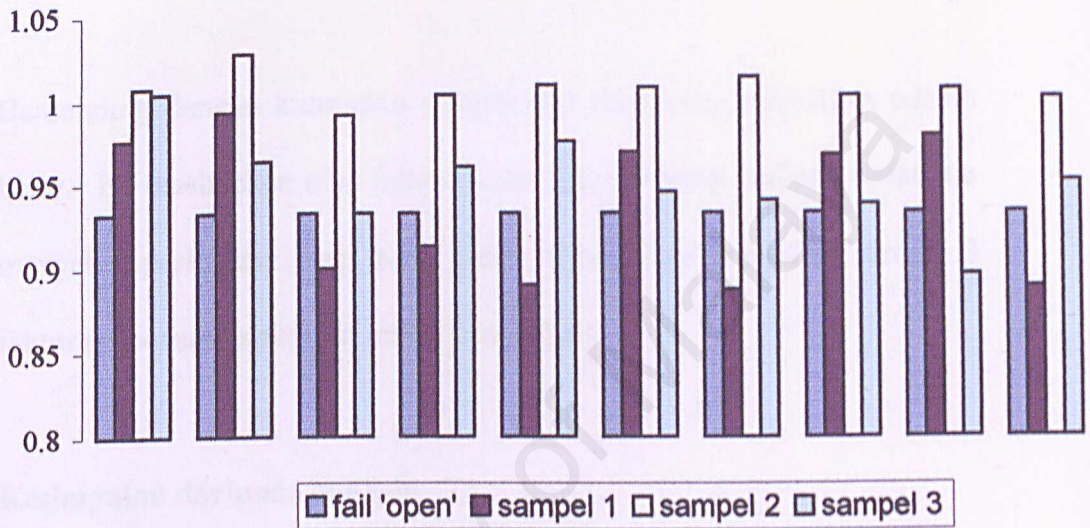
Jadual 4.4 : purata ralat bagi ketiga tiga kumpulan sampel

Nilai-nilai ini digambarkan dalam bentuk peratusan seperti yang ditunjukkan dalam carta pai di bawah



Rajah 4.7: Peratusan Ralat yang wujud dalam ketiga-tiga kumpulan Sampel

Daripada carta pai di atas di dapati bahawa sampel dari kumpulan 1 mempunyai purata ralat yang paling kecil diikuti oleh kumpulan sampel ketiga manakala purata ralat yang paling tinggi ditunjukkan dalam sampel data kumpulan kedua. Keadaan ini boleh digambarkan dengan memplotkan graf seperti di bawah :



Rajah 4.8 : Perbandingan purata nilai (\bar{x}) bagi ketiga tiga kumpulan sampel

Secara umumnya setiap individu mempunyai cara atau gaya percakapan yang berbeza yang mewujudkan keunikan dalam suara setiap individu. Selain itu jenis perkataan yang disebut juga memainkan peranan penting kerana setiap perkataan adakalanya disebut dengan kadar yang berbeza. Ini mewujudkan spectrum frekuensi yang berbeza bagi setiap isyarat mentah yang digunakan.

Kewujudan ralat dalam sampel serta nilai unik bagi setiap sampel dipengaruhi oleh faktor ini. Kumpulan sampel 1 mempunyai ralat yang paling sedikit memandangkan setiap rekod suara yang dilakukan adalah dari orang yang sama serta perkataan yang sama mewujudkan spektrum frekuensi yang hampir sama menjadikan nilai (x) yang dihasilkan adalah hampir sama.

Berbanding dengan kumpulan sampel dua ralat yang dihasilkan adalah besar. Ini disebabkan oleh faktor suara individu yang berbeza walaupun menyebut perkataan yang sama. Sebaliknya dalam kumpulan sampel 3 faktor perbezaan perkataan lebih ditonjolkan.

4.2.5 Kesimpulan daripada analisis

Kumpulan sampel 1 diwujudkan untuk melihat sejauh mana keberkesanan kod yang dihasilkan dan ternyata penggunaan sampel seperti ini (suara dan perkataan yang sama) adalah lebih berkesan walaupun masih menghasilkan ralat.

Kumpulan sampel kedua diwujudkan untuk melihat pengaruh perbezaan di antara suara individu berlainan. Apa yang diperolehi adalah, faktor ini menghasilkan ralat yang lebih besar menunjukkan pengaruh yang lebih banyak ke atas nilai nilai (x) yang dihasilkan.

Kumpulan sampel 3 pula diwujudkan untuk mengkaji pengaruh perbezaan perkataan yang disebut. Hasilnya didapati ralat yang diperolehi adalah lebih kecil berbanding kumpulan sampel 2 menunjukkan faktor ini mempunyai pengaruh yang lebih kecil ke atas nilai data. Dengan kata lain suara individu yang berlainan akan menghasilkan ralat yang lebih besar berbanding dengan pemilihan perkataan yang berlainan.

4.3 RINGKASAN

Secara keseluruhannya bab ini membincangkan tentang nilai unik yang diperolehi bagi setiap kumpulan sampel serta membuat penilaian ke atas nilai tersebut di samping mengenal pasti keunikan di sebalik nilai unik tersebut. Sebarang perbezaan yang wujud dalam setiap kumpulan sampel dikaji. Penggunaan carta pai dan graf dipilih bagi memudahkan analisis yang dilakukan.

Walaupun setiap individu mempunyai suara yang berlainan tetapi untuk mengesan keunikan yang ada bukanlah satu perkara yang mudah. Masalah serta faktor yang memungkinkan keadaan ini dibincangkan dalam bab seterusnya.

Dari pada halaguna anafis di dapan balat ada mada. Dapa yang
kumpulan mada. Balat ada mada mada yang mada mada
mada mada mada mada mada mada mada mada mada mada mada
mada mada mada mada mada mada mada mada mada mada mada
yang mada mada mada mada mada mada mada mada mada mada

19

Untuk mada mada mada mada mada mada mada mada mada mada

lagi mada mada mada mada mada mada mada mada mada mada

mada mada mada mada mada mada mada mada mada mada

Bab 5

Perbincangan

101	12.000	12.000
102	12.000	12.000
103	12.000	12.000
104	12.000	12.000
105	12.000	12.000
106	12.000	12.000
107	12.000	12.000
108	12.000	12.000
109	12.000	12.000
110	12.000	12.000
111	12.000	12.000
112	12.000	12.000
113	12.000	12.000
114	12.000	12.000
115	12.000	12.000
116	12.000	12.000
117	12.000	12.000
118	12.000	12.000
119	12.000	12.000
120	12.000	12.000

Jadual 1.1 : Perbincangan mada mada mada mada mada mada mada mada mada mada

mada mada mada mada mada mada mada mada mada mada mada mada mada

5.1 FAKTOR YANG MEMUNGKINKAN RALAT

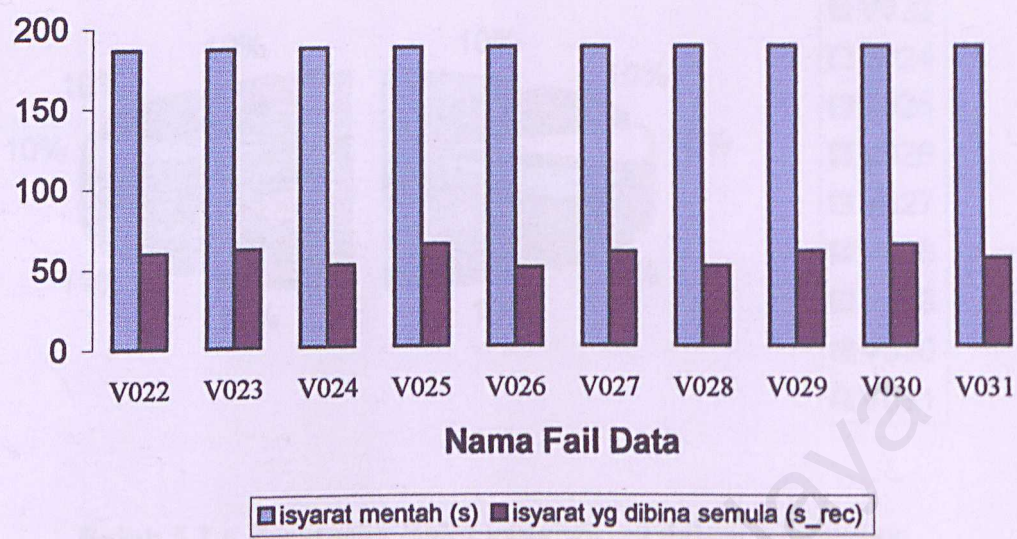
Daripada bahagian analisis di dapati bahawa ralat muncul dalam setiap kumpulan sampel. Beberapa factor telah dikenalpasti yang berpotensi menjadi penyebab wujudnya ralat-ralat ini. Dari segi srtuktur aturcara, masalah timbul ketika proses penyahkomposisian dan pembinaan semula apabila nilai isyarat yang dibina semula (s_{rec}) jauh lebih rendah berbnading nilai isyarat mentah (s).

Untuk mengesahkan keadaan ini, analisis dilakukan ke atas nilai s dan s_{rec} bagi setiap sampel data dalam setiap kumpulan seperti mana yang dilakukan ke atas nilai unik (x). Hasilnya adalah seperti yang dipaparkan dalam jadual di bawah :

Nama fail	Purata nilai (s)	Purata nilai (s_{rec})	Purata nilai ralat
V022	186.6106	59.6118	126.9988
V023	186.6106	61.3656	125.2449
V024	186.6106	51.0765	135.5341
V025	186.6106	63.5310	123.0795
V026	186.6106	49.0767	137.5339
V027	186.6106	58.6757	127.9358
V028	186.6106	49.7081	136.9025
V029	186.6106	58.4200	128.1907
V030	186.6106	62.5112	124.0993
V031	186.6106	54.7655	131.8450
Purata nilai ralat untuk 10 sampel			129.7364

Jadual 5.1 : Perbezaan antara nilai s dan s_{rec} dalam sampel kumpulan satu Serta purata nilai ralat yang muncul.

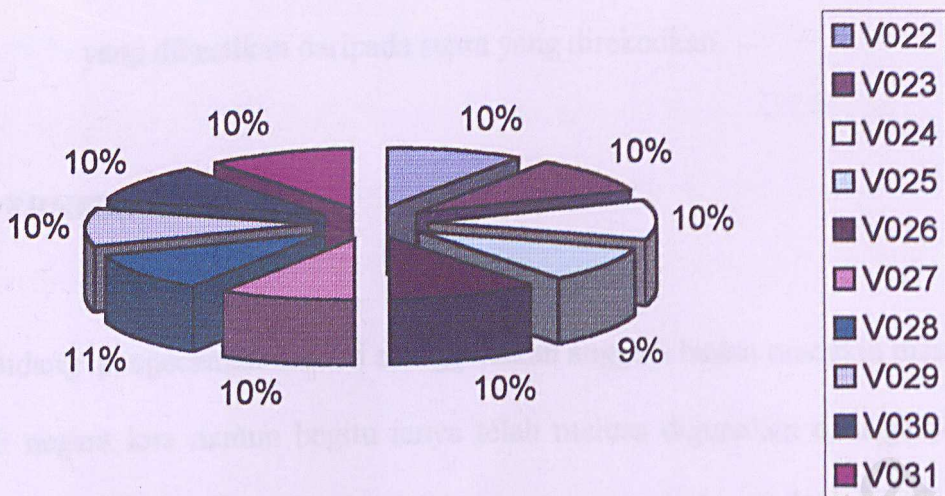
Perbezaan ini dapat dilihat dengan lebih jelas berdasarkan graf dalam rajah berikut:



Rajah 5.1 : Perbandingan purata nilai (s) dan (s_rec) bagi setiap sampel dalam kumpulan satu.

Daripada graf dapat dilihat bahawa perbezaan yang wujud adalah amat ketara menghasilkan nilai purata keseluruhan ralat adalah 129.7364. Masalah ini turut berlaku bagi kumpulan sampel dua dan tiga. Ralat yang dihasilkan oleh kedua-dua kumpulan sampel ini adalah hampir sama dengan kumpulan sampel 1. Ini bermakna proses penyahkomposisian dan pembinaan semula turut mengalami ralat dalam kesemua sampel yang ada.

Peratusan ralat yang wujud dalam sampel kumpulan satu di gambarkan melalui carta pai di bawah :



**Rajah 5.2 : Peratusan Ralat yang wujud dalam kumpulan
Sampel 1**

Selain itu terdapat beberapa faktor lain yang turut dikenalpasti menjadi punca wujudnya ralat dalam sampel. Antara faktor tersebut termasuklah :

- i. Suara manusia yang biasanya mudah dipengaruhi oleh banyak faktor terutamanya emosi. Perasaan manusia apabila marah, sedih gembira dan sebagainya menghasilkan nada suara yang berbeza-beza sekalipun orang yang sama. Keadaan ini menyukarkan suara direkod pada normal untuk menghasilkan isyarat yang sama.

- ii. Faktor persekitaran juga tidak kurang pentingnya. Keadaan persekitaran yang bising sedikit sebanyak boleh mempengaruhi isyarat yang dihasilkan daripada suara yang direkodkan.

5.2 PERKEMBANGAN

Bidang pengecaman identiti menggunakan anggota badan mungkin masih baru di negara kita namun begitu ianya telah meluas digunakan di negara-negara barat. Kebanyakannya menggunakan retina, cap jari, wajah dan sebagainya. Walau bagaimanapun, dalam bidang pengecaman suara kajian demi kajian masih lagi giat dijalankan bagi mendapatkan cara terbaik mengenal identiti manusia menggunakan suara.

Ini adalah kerana keunikan yang tersembunyi dalam suara setiap individu amat sukar dikenalpasti kerana ianya amat sensitif dengan banyak faktor seperti yang dinyatakan dalam bahagian sebelumnya. Keadaan menjadi lebih sukar untuk membina sistem pengecaman suara individu tertentu yang bebas menyebut apa sahaja perkataan berbanding dengan menyebut perkataan yang telah dikhaskan.

Namun begitu ianya boleh dilakukan sekiranya faktor-faktor yang boleh menyebabkan ralat dititikberatkan. Suara-suara direkodkan pada keadaan yang normal. Selain itu teknik normalisasi juga membantu memudahkan pengenalanpastian keunikan dalam suara manusia.

Sebagaimana sukarnya mengenalpasti keunikan suara seperti itu jugalah keberkesanan penggunaan nilai keunikan ini. Teknologi pengecaman suara memungkinkan wujudnya satu sistem pertahanan yang kukuh. Semakin unik nilai yang diperolehi semakin kukuh sistem yang di bina. Ini kerana jarang sekali terdapat manusia yang dapat meniru suara orang lain dengan tepat.

Dengan penggunaan teknologi teknologi serta peralatan yang lebih canggih dan jitu, tidak mustahil teknologi pengecaman suara ini akan lebih berkesan dan digunakan secara meluas di seluruh dunia.

5.3 RINGKASAN

Secara keseluruhannya kod yang telah dihasilkan dalam projek ini sedikit sebanyak telah menepati ciri-ciri pengecaman suara. Walau bagaimanapun masih terdapat faktor-faktor yang perlu di titikberatkan bagi menghapuskan ralat yang wujud dalam aturcara.

Dari analisis yang dilakukan kesilapan mungkin muncul dari aturcara ataupun faktor luaran semasa pembentukan fail data. Kajian terperinci atau peralatan serta teknik yang canggih mungkin akan memudahkan projek seumpama ini di masa hadapan terutamanya dalam usaha mendapatkan data yang lebih tepat.

- [1] Jessi Allen (1978). *Web Developer's Own Guide To Streamline Multimedia*. John Wiley & Sons Inc.
- [2] William B. Gervase (1994). *Artificial Intelligence Expert Systems Companion: Vision and Natural Language Processing*. Hayes Publications.
- [3] Rajala De Niro Sam Ching Y. (2000). *New Systems and Architectures for Automatic Speech Recognition and Speech Synthesis*. World Scientific.
- [4] Yihui Zhang's for John D. (2000). *Speech Recognition and Synthesis: A Modern Approach*. Morgan Kaufmann.
- [5] Kenneth Jensen (1997). *Speech Processing*. McGraw-Hill.
- [6] John C. (2000). *Robust Speech Recognition in Embedded Systems and Its Applications*. Kluwer Academic Publishers.
- [7] Zaidi Faris (2000). *Perkembangan YML: Pengaruh Terhadap Perkembangan Kias*. Penerbitan Institut Warisan, Medan Timur, Johor Bahru, Malaysia.

Rujukan

RUJUKAN

- [1] Jose' Alvea (1968). Web Developer.Com Guide To Streaming Multimedia. John Wiley & Sons Inc.
- [2] William B. Gevarter (1984). Artifial Intelligence Expert Systems Computer Vision and Natural Language Processing. Noyes Publications.
- [3] Renato De Mori dan Ching Y. Suen. New Systems and Architectures For Automatic Speech Recognition and Speech Synthesis. NATO Scientific Affairs Division.
- [4] Vinay K. Ingle dan John G. Proakis (2000). Digital Signal Processing Using Matlab. Thompson Learning.
- [5] Kermit Sigmon (1998). Matlab Primer. 5th ed. CRC Press.
- [6] Jean-Claude Junqua (2000). Robust Speech Recognition in Embedded Systems and PC Applications. Kluwer Academic Publishers.
- [7] Zaidi Razak (2000). Perlaksanaan VHDL Bagi Rekabentuk Perkakasan Khas Penukarganti Diskrit Wavelet. Master Thesis. Universiti Malaya.

- [8] Jaideva C. Goswami, Andrew K. Chan (1999). Fundamentals of wavelets: Theory, Algorithms and Application. John Wiley and Sons Inc.
- [9] Azree Idris (1999). Matlab for Engineering Students. Prentice Hall (M) Sdn. Bhd.
- [10] <http://www.dv.co.yu/mpgscript/mpeghdr.htm>
- [11] http://mpegdit.org/mpegdit/mpeg_format/MP3Format.html
- [12] <http://howstffwork.lycoszone.com/MP31.htm.htm>
- [13] <http://www.wikipedia.org/wiki/MP3>
- [14] <http://www.newint.org/headphones/format.htm>
- [15] <http://www.my-mp3-player.co.uk/mp3glossary.html>
- [16] <http://www.sonicspot.com/guide/fileformatlist.html>
- [17] http://www.audiofilesolutions.com/comparison_table.htm
- [18] <http://www.borg.com/~jglatt/tech/aiff.htm>

- [19] <http://astronomy.swin.edu.au/~pbourke/dataformats/aiff>
- [20] <http://www.sonicspot.com/guide/wavefiles.html>
- [21] <http://www.borg.com/~jglatt/tech/wave.htm>
- [22] <http://astronomy.swin.edu.au/~pbourke/dataformats/wave>
- [23] <http://www.tsp.ece-mcgill.ca/mmsp/Documents/AudioFormats>
- [24] <http://www.sonicspot.com/guide/midifiles.html>
- [25] <http://www.borg.com/~jglatt/tech/midifiles.htm>
- [26] http://crystal.apana.org.au/ghansper/midi_introduction/midi_file_format.html
- [27] http://ourworld.compuserve.com/homepages/mark_clay/midi.htm
- [28] <http://www.dazyweblabs.com/wavinfo>
- [29] <http://www.neurotraces.com/scitab/scilab2/nnode24.htm>

- [30] <http://sharkysoft.com/software/java/lava/docs/javadoc/lava/rift/wave/docfiles/riffwave-content.htm>
- [31] <http://www.microsoft.com/windows/windowsmedia/format/asfspec.aspx>
- [32] <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmform/htm/overviewofmeasfformat.asp>
- [33] <http://www.borg.com/~jglatt/tech/wave.htm>
- [34] <http://ccrma-www.stanford.edu/ccrma/courses/422/projects>
- [35] <http://www.cs.tut.fi/sgn/arg/kujahalk/speaker>
- [36] http://www.speech.kth.se/~rolf7gslt_papers/SvetoslavMarinov.pdf
- [37] <http://mi.eng.cam.ac.uk/comp.speech/Section6/Q6,1.html>
- [38] <http://cslu.cse.ogi.edu/HLTSurvey/chlnode9.html>
- [39] <http://www.cse.iitd.emet-in/~csu00108/report/repon.html>

LAMPIRAN 4

Pengalaman dalam Kegiatan Pembelajaran

Prinsip

- Rencanakan pembelajaran
- Dams (Data dan Informasi) yang relevan
- Jelaskan dan jelaskan
- Perhatikan dan perhatikan
- Perhatikan dan perhatikan
- Perhatikan dan perhatikan
- Perhatikan dan perhatikan
- Perhatikan dan perhatikan

Ked

Lampiran

LAMPIRAN A

Pengecaman suara menggunakan Matlab.

Proses:

- Rekod fail wav
- Baca fail wav menggunakan wavread
- Jalankan Fast Fourier Transform
- Wakilkan nombor dalam matrik FFT sebagai nombor nyata dengan penbaraban dengan konjugatnya
- Dapatkan nilai yang tepat bagi data dalam matrik yang baru
- Pecahkan matrik FFT ke dalam n bins dan dapatkan nilai purata
- Matrik dibahagikan dengan jumlah data yang dikembalikan.

Kod

Semua kod yang digunakan ditulis dalam Matlab. Terdapat dua aturcara Matlab yang digunakan iaitu SoundSig.m dan juga run.m untuk tujuan perbandingan.

```
SoundSig.m : function [p]=soundSig(filename)
               in=wavread(filename);
               f=fft(in,100000);
               q=f.*conj(f);
               q=abs(q(1:5000));
               for i=1:4
                   t=(i-1)*100+1;
                   p(i)=sum(q(t:t+99));
               end
               p=p/sum(p);
```



```
run.m : test=soundSig('mewave');
        alok=soundSig('alokwave');
        fugi1=soundSig('meagainwave');
        fugi2=soundSig('meagain2wave');
        jon=soundSig('jonwave');
        alokother=soundSig('alok2wave');
        fugiother=soundSig('me2wave');
```

```
sum(test.*alok)
sum(test.*fugi1)
sum(test.*fugi2)
sum(test.*jon)
sum(test.*alokother)
sum(test.*fugiother)
```

Hasil :

7 fail wav dari 3 orang yang berlainan digunakan untuk dibuat perbandingan.

Nilai yang lebih tinggi menunjukkan suara yang hampir sama.

Subjek : Jonathan Terleski, Alok Lasdariya, Jonathan chu

Terleski-1 Vs. Ladsariya-1	0.0266
Terleski-1 Vs. Terleski-2	0.1775
Terleski-1 Vs. Terleski-3	0.2083
Terleski-1 Vs. Ladsariya-2	0.0135
Terleski-1 Vs. Terleski-4	0.0900
Terleski-1 Vs. Chu-1	0.0219
Ladsariya-1 Vs. Chu-1	0.1016

Daripada jadual di atas, didapati bahawa Terleski -N adalah berpadanan antara satu sama lain (mempunyai nilai yang tinggi apabila Terleski dibandingkan dengan Terleski yang lain). Terleski-1,2 dan 3 adalah sebutan perkataan yang sama. Nilai bagi Terleski 4 agak berbeza apabila dibandingkan. Ini kerana perkataan yang disebut adalah berlainan dan merupakan soalan. Peningkatan frekuensi berlaku apabila ayat yang disebut merupakan soalan.

Kesimpulan :

Proses di atas menerangkan secara terperinci mengenai langkah-langkah untuk pengecaman suara. Hasil menunjukkan ianya adalah tidak mustahil untuk mengesan suara bagi individu tertentu.

Walau bagaimanapun, ianya terhad kepada kekangan tertentu seperti ianya bukanlah sesuatu yang masa nyata. Untuk melakukannya kita perlu merekod suara dalam bentuk fail wav dan melarikan fail mengikut aturcara di atas.

LAMPIRAN B

- **rejected.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 39231

- **enter system.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 53247

- **accepted.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 47644

- **hello.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 33630

- **buka.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 39231

- **tutup.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 39231

- **suara.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 36432

- **masuk.wav**

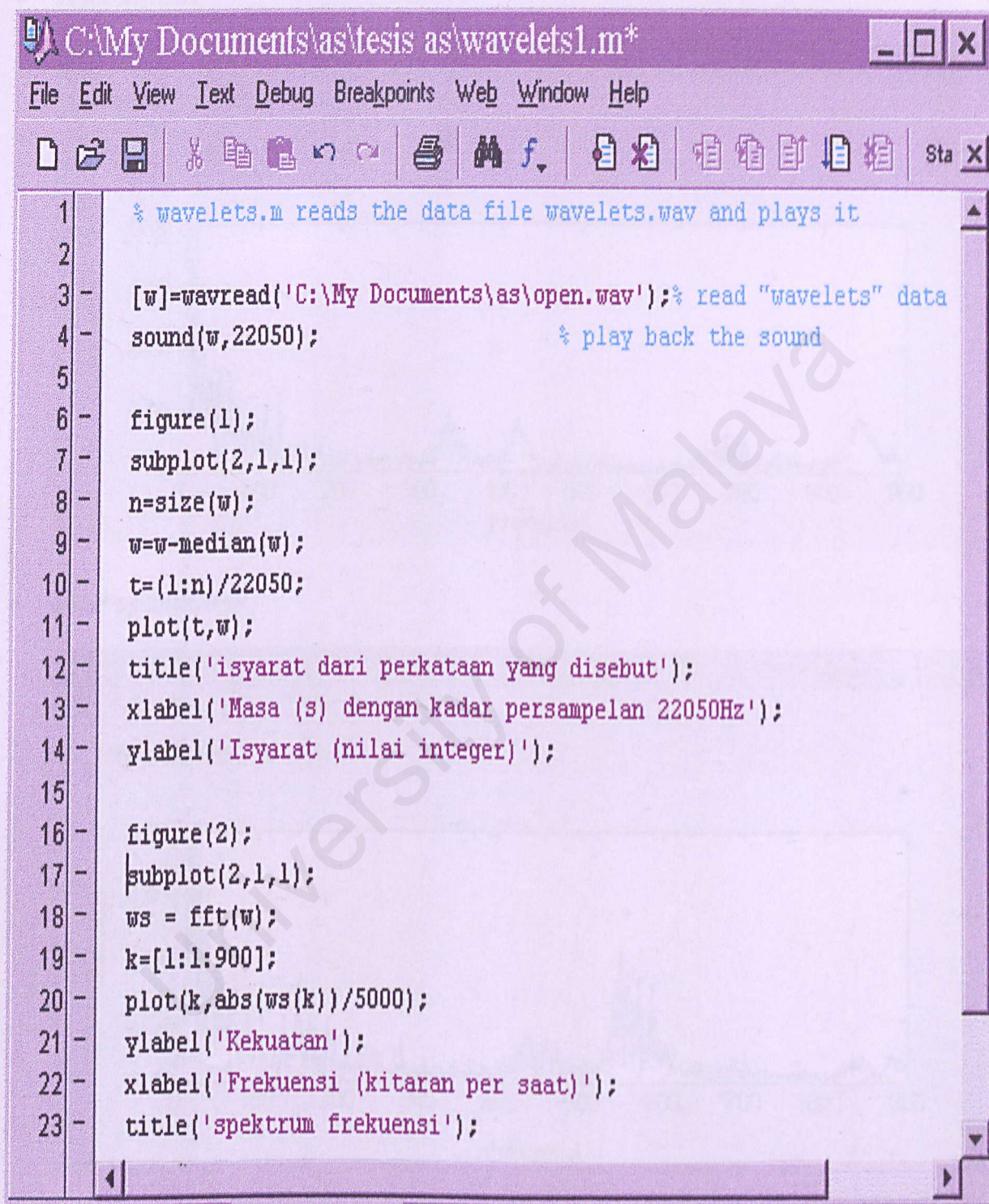
FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 39231

- **bunyi.wav**

FormatTag: 1
FormatDescription: 'Microsoft WAVE_FORMAT_PCM'
NumberOfChannels: 1
SampleRate: 22050
AvBytesPerSec: 22050
BlockAlign: 1
BitsPerSample: 8
ContainerBytes: 1
ContainerBits: 8
SamplesPerChannel: 39231

LAMPIRAN C

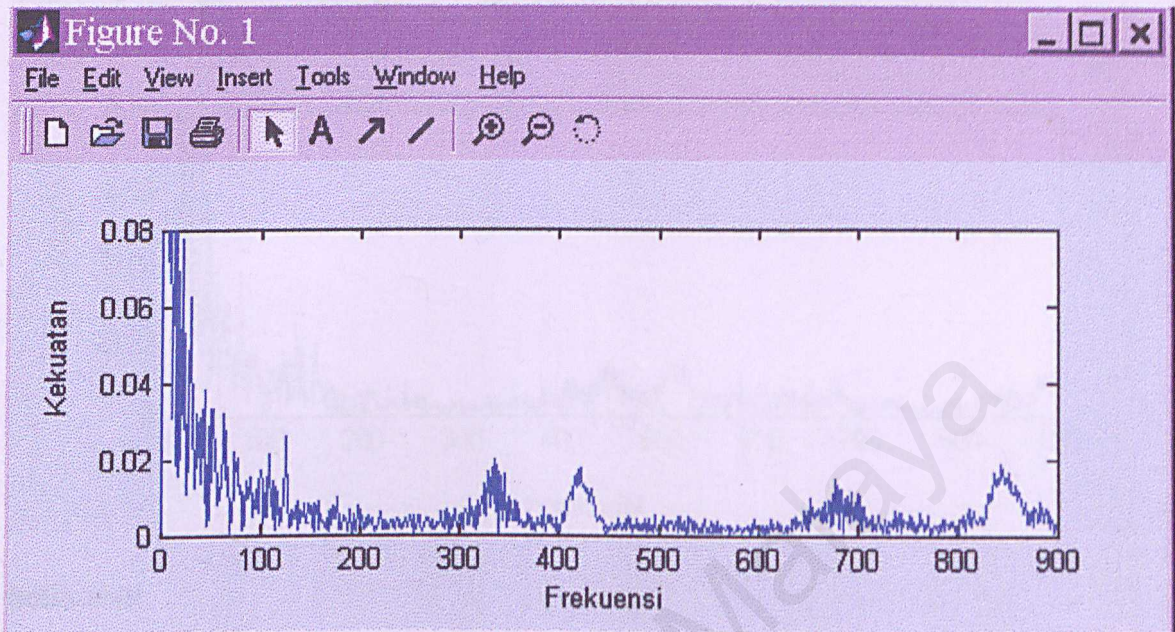
Aturcara yang digunakan untuk mendapatkan speltrum frekuensi.



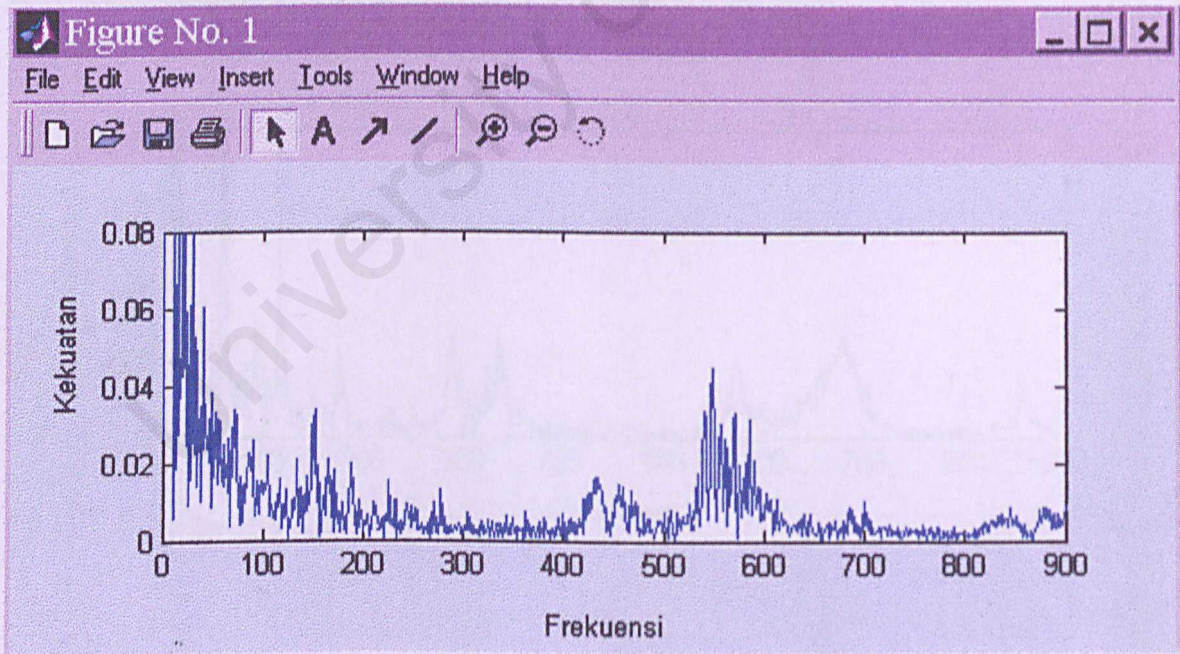
```
1 % wavelets.m reads the data file wavelets.wav and plays it
2
3 [w]=wavread('C:\My Documents\as\open.wav');% read "wavelets" data
4 sound(w,22050); % play back the sound
5
6 figure(1);
7 subplot(2,1,1);
8 n=size(w);
9 w=w-median(w);
10 t=(1:n)/22050;
11 plot(t,w);
12 title('isyarat dari perkataan yang disebut');
13 xlabel('Masa (s) dengan kadar persampelan 22050Hz');
14 ylabel('Isyarat (nilai integer)');
15
16 figure(2);
17 subplot(2,1,1);
18 ws = fft(w);
19 k=[1:1:900];
20 plot(k,abs(ws(k))/5000);
21 ylabel('Kekuatan');
22 xlabel('Frekuensi (kitaran per saat)');
23 title('spektrum frekuensi');
```


Spektrum frekuensi bagi fail-fail data yang lain:

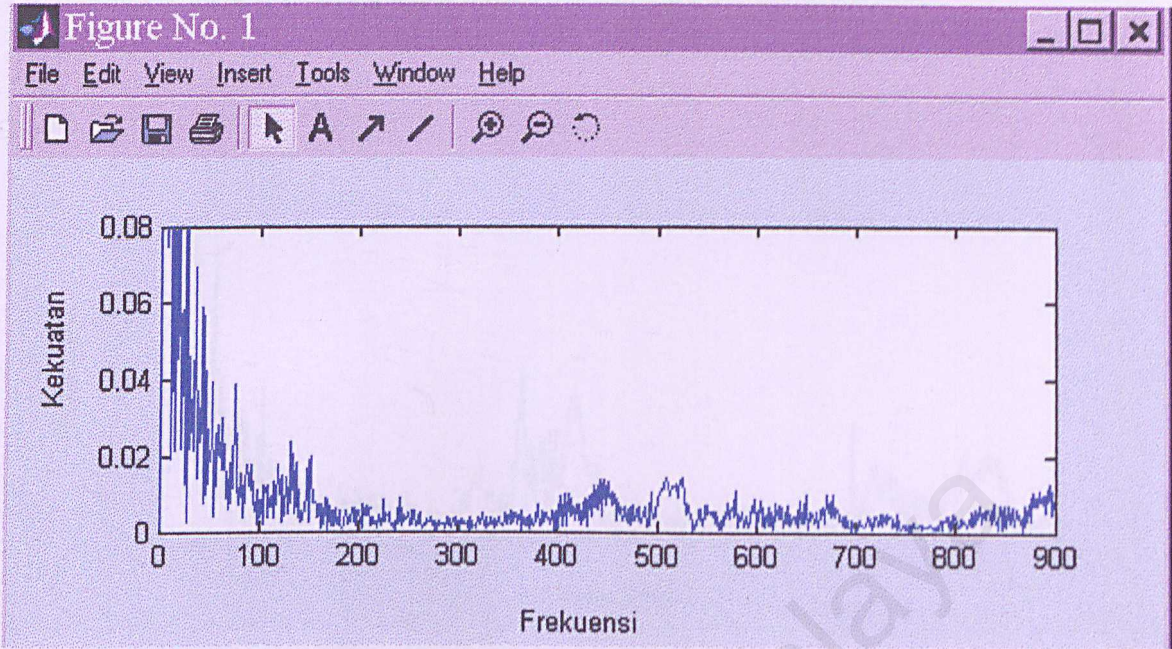
- rejected.wav



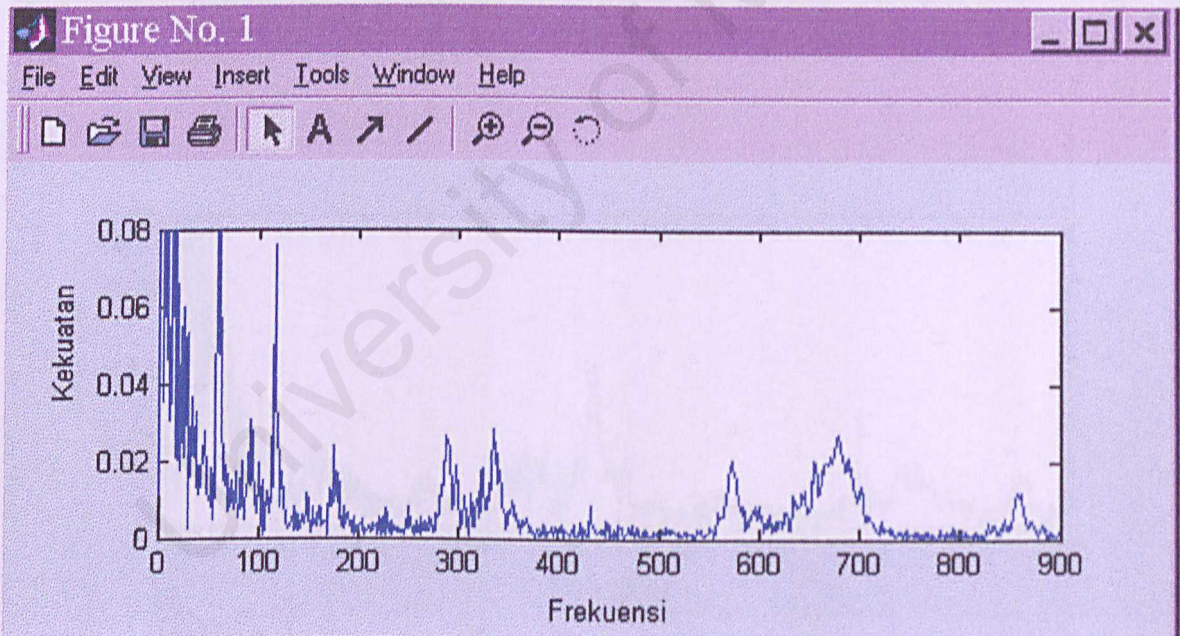
- enter system.wav



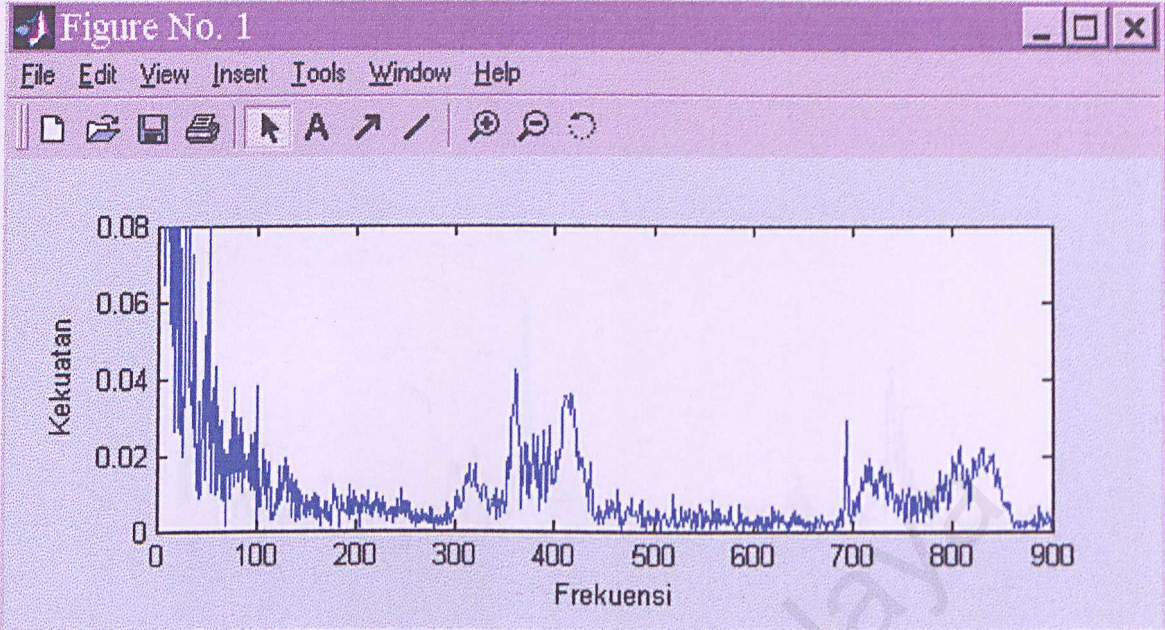
- **accepted.wav**



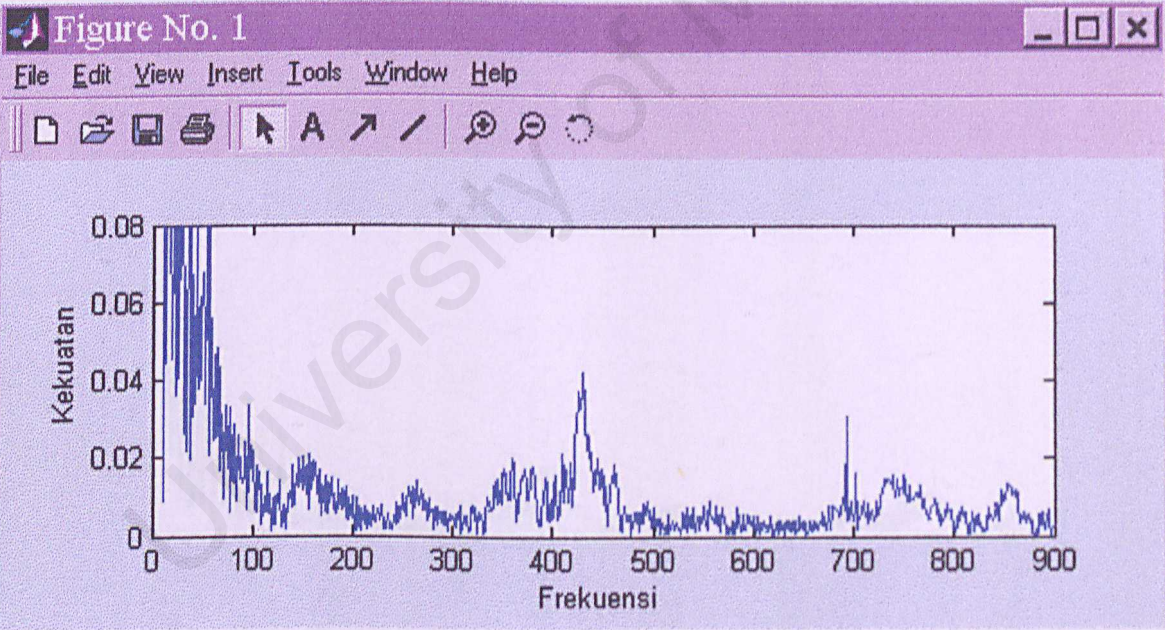
- **hello.wav**



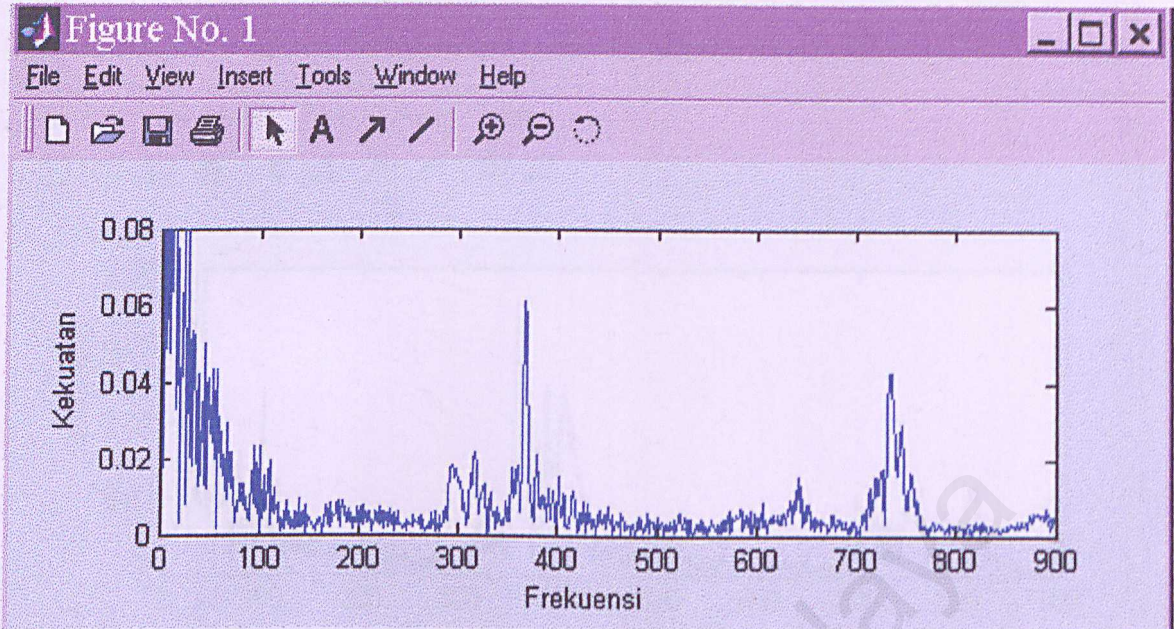
- buka.wav



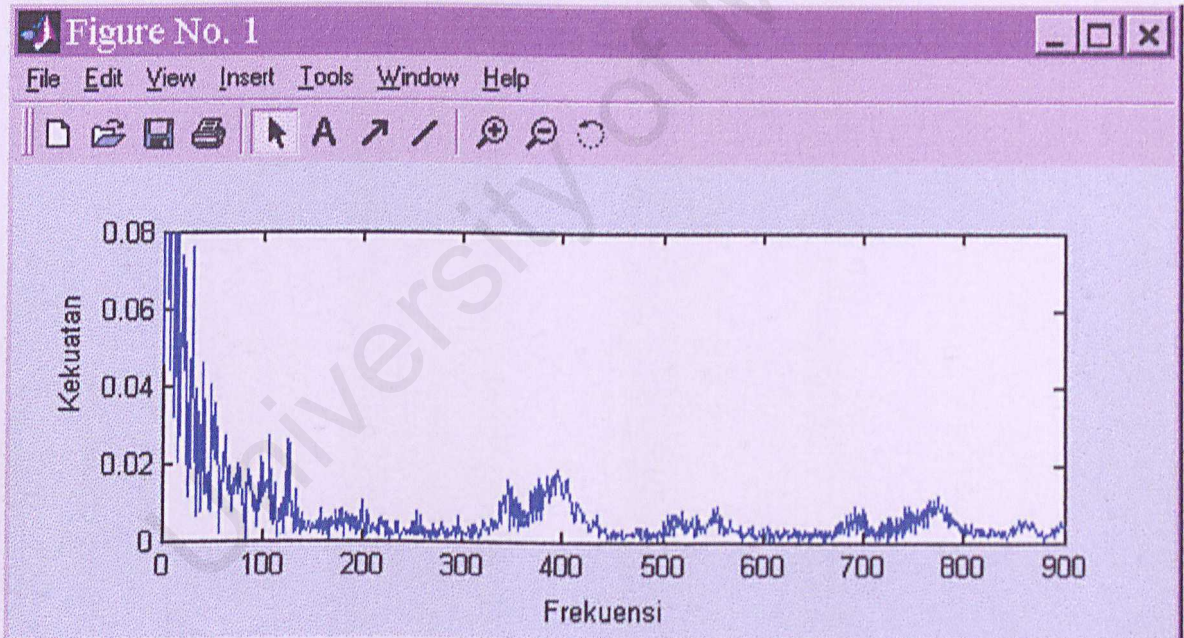
- tutup.wav



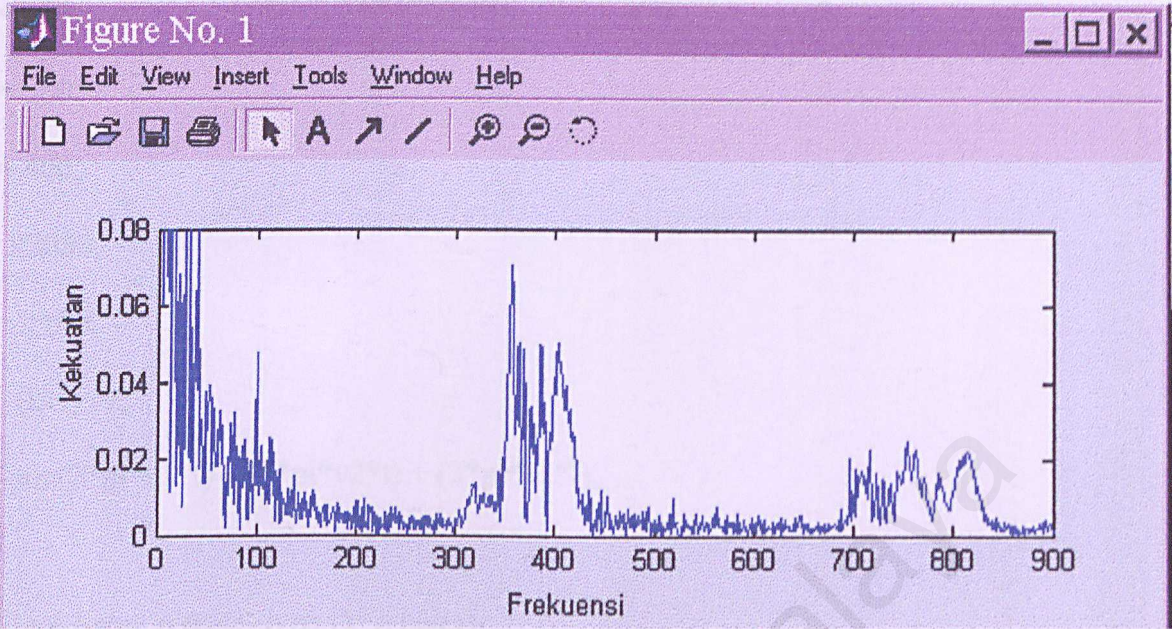
- **suara.wav**



- **masuk.wav**



- bunyi.wav



LAMPIRAN D

$$v1 = 100;$$

$$v2 = 200;$$

$$v3 = 400;$$

$$r = 22050;$$

$$k = 1:100;$$

$$t = (k-1) / r;$$

$$s = \sin (2*\pi*v1*t) + (2*\pi*v2*t) + (2*\pi*v3*t);$$

$$g0 = [0.9505; 0.0351; 0.0111; 0.0032];$$

$$k = [0; 1; 2; 3];$$

$$g1 = \text{flipud}(g0).*(-1).^k;$$

$$h0 = \text{flipud}(g0) / 2;$$

$$h1 = \text{flipud}(g1) / 2;$$

$$x = \text{conv}(s,h0);$$

$$a0 = x(1:2:\text{length}(x));$$

$$x = \text{conv}(s,h1);$$

$$w0 = x(1:2:\text{length}(x));$$


```
y = zeros (2*length (w0),1);  
y(1:2:2*length (w0)) = w0 (1: length (w0));  
x = zeros (2*length (a0_rec),1);  
x(1:2:2*length (a0_rec)) = a0_rec;  
x = conv (x,g0)  
y = conv (y,g1);  
y = x(1:length(y)) + y;  
s_rec = y (4:length(y)-4);
```

figure (2)

```
subplot (3,2,1), plot (a1)
```

```
ylabel ('a_{-1}')
```

```
subplot (3,2,2), plot (w1)
```

```
ylabel ('w_{-1}')
```

```
subplot (3,2,3), plot (a0_rec)
```

```
ylabel ('reconstructed a_0')
```

```
subplot (3,2,4), plot (w0)
```

```
ylabel ('w_0')
```

```
subplot (3,2,5), plot (s_rec)
```

```
ylabel ('reconstructed signal')
```

```
set (gcf, 'paperposition', [0.5 0.5 7.5 10])
```

```
x = conv (a0,h0);
```

```
a1 = x(1:2:length (x));
```

```
x = conv (a0,h1);
```

```
w1 = x(1:2:length (x));
```

```
subplot (3,2,1), plot (s)
```

```
ylabel ('signal')
```

```
subplot (3,2,3), plot (a0)
```

```
ylabel ('a_0')
```

```
subplot (3,2,4), plot (w0)
```

```
ylabel ('w_0')
```

```
subplot (3,2,5), plot (a1)
```

```
ylabel ('a_{-1}')
```

```
subplot (3,2,6), plot (w1)
```

```
ylabel ('w_{-1}')
```

```
set (gcf, 'paperposition', [0.5 0.5 7.5 10])
```

```
x = zeros (2*length (a1),1);
```

```
x(1:2:2*length (a1)) = a1(1: length (a1));
```

```
y = zeros (2*length (w1),1);
```

```
y(1:2:2*length (w1)) = w1 (1: length (w1));
```

```
x = conv (x,g0) + conv (y,g1);
```

```
a0_rec = x (4:length(x)-4);
```


LAMPIRAN E

Aturcara yang digunakan untuk mengira ralat yang wujud dalam setiap sample yang di analisis.

```
#include<stdio.h>

#include<math.h>

void main() {
float data_a[110],data_b[110],beza[110],purata,jumlah;
int i=0;
FILE *fi_1,*fi_2,*fo;

fi_1=fopen("a1.txt","r");
fi_2=fopen("b1.txt","r");
while (!feof(fi_1))
    {
        fscanf(fi_1,"%f",&data_a[i]);
        printf("%f\n",data_a[i]);
        i++;
    }
fclose(fi_1);
printf("\n\n");

i=0;
while (!feof(fi_2))
    {
        fscanf(fi_2,"%f",&data_b[i]);
        printf("%f\n",data_b[i]);
        i++;
    }
```

```
fclose(fi_2);
```

```
printf("\n\n");
```

```
i=0;
```

```
fo=fopen("c1.txt","w");
```

```
for (i=0;i<100;i++)
```

```
{
```

```
    beza[i]=data_a[i]-data_b[i];
```

```
    printf("%f\n",fabs(beza[i]));
```

```
    fprintf(fo,"%f\n",fabs(beza[i]));
```

```
}
```

```
i=0;
```

```
jumlah=0.0;
```

```
for (i=0;i<100;i++)
```

```
    jumlah=jumlah+fabs(beza[i]);
```

```
    purata=jumlah/100;
```

```
    printf("\nPurata= %f",purata);
```

```
    fprintf(fo,"\nPurata= %f",purata);
```

```
fclose(fo);
```

```
}
```